CPExpert™

WLM Component Volume 2

Computer Management Sciences, Inc
6076-D Franconia Road
Alexandria, Virginia 22310-1756
voice: (703) 922-7027
fax: (703) 922-7305
www.cpexpert.com

CPExpert is a trademark of Computer Management Sciences, Inc.

This manual applies to the WLM Component of **CPExpert**[tm], a proprietary software product of Computer Management Sciences, Inc., Alexandria, Virginia, United States of America.

The information in this document is subject to change. Comments or suggestions are welcome, and to the extent practicable, will be incorporated in revisions to this document. Please send all comments, suggested new rules, suggested changes to existing rules, suggestions for improvement to the software or documentation, or any other advice to:

Computer Management Sciences, Inc.
6076-D Franconia Road
Alexandria, Virginia 22310-1756
(703) 922-7027 FAX: (703) 922-7305
www.cpexpert.com

DISCLAIMER

The advice, recommendations, or otherwise contained in this document represent information generally available in the public domain, as contained in vendor manuals, published in articles or papers, presented at professional conferences, or otherwise commonly accepted in the professional community. Neither Computer Management Sciences, Inc. nor its representatives make representations or warranties with respect to the applicability or application of any advice, recommendations, or otherwise, contained in this document or in any results from applying the CPExpert software, to any particular computer system or computer installation.

TRADEMARKS

CPExpert is a trademark of Computer Management Sciences, Inc. IBM, MVS/370, MVS/SP, MVS/XA, MVS/ESA, Enterprise System/3090, Netview, PR/SM, Processor Resource/System Manager, Hiperspace, and ES/9000 are trademarks of the IBM Corporation. SAS, SAS/OR, and SAS/STAT are trademarks of the SAS Institute Inc. MXG is a trademark of Merrill Consultants. MICS is a trademark of Legent Corporation.

COPYRIGHT INFORMATION

©Copyright 1994, Computer Management Sciences, Inc.
All rights reserved. Printed in the United States of America.

This licensed work is confidential and propriety, and is the property of Computer Management Sciences, Inc. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the specific written authorization of Computer Management Sciences, Inc.

Description of Rules

Appendix A

©Copyright 1994, Computer Management Sciences, Inc.

This appendix contains a description of each rule that results in a finding by the WLM Component of CPExpert. The description summarizes the rule, lists predecessor rules, discusses the rationale for the finding, and suggests action. The Appendix is contained in both Volume 1 and Volume 2 of this User Manual.

The summary of the rule presents a short description of the finding.

The predecessor rules are listed so you can follow the line of reasoning leading to a particular rule being executed.

The discussion describes as much as necessary of the operation of the computer system (the hardware, the WLM, the SRM, etc.) as it relates to the particular rule. The purpose of the discussion is to explain the reasoning behind the rule, and what causes the rule to be produced.

The suggestions list possible actions that should be considered based on the findings. In many cases, multiple possible actions are listed. You must determine which actions should be taken (this determination is based upon the suitability of the actions to your own environment, the financial implications of the action, and the "political" acceptability of the action.)

The rules are organized in numerical order. However, not all numbers are represented (for example, RULE WLM200 follows RULE WLM150). The LIST OF RULES in this appendix lists all rules that are included in the initial release of the WLM Component. Within the rule framework, the following general categories apply:

- **Service Policy Findings**. The Service Policy Findings are rules in the WLM001 to WLM050 range. These findings help identify problems or potential problems with the Workload Manager service definition. The Service Policy Findings are contained in Volume 1.

  It is important to realize that these findings normally identify a POTENTIAL problem. Your systems programming staff must decide whether the findings (and their associated recommendations) make sense in your environment. For example, your systems programming staff might have deliberately selected certain parameter values. The values might be appropriate for your installation and your management objectives, even though CPExpert might produce a rule indicating that there is a potential problem with the parameter.

  You can disable CPExpert's checking the service definition by modifying the CHKPLCY guidance variable in USOURCE(WLMGUIDE). If the CHKPLCY guidance variable is set to N, CPExpert will not check the service definition for potential problems.

©Copyright 1994, Computer Management Sciences, Inc.

- **General System Findings**. The General System Findings are rules in the WLM050 to WLM099 range. These findings identify problems or potential problems with your overall system. For example, many of the rules deal with problems with the paging subsystem. These findings are made only if CPExpert detected that a performance goal was not met and that some general system problem might have caused the goal to be missed. The General System Findings are contained in Volume 1.

- **Specific Findings**. The Specific Findings are rules above WLM100. These findings are made if CPExpert detected that a service class did not meet its performance goal. In the Specific Findings, CPExpert attempts to isolate the reason(s) the performance goal was not met. The Specific Findings are contained in Volume 2.

  WLM1nn(series) relate to performance goal findings

  WLM2nn(series) relate to CPU-related findings

  WLM3nn(series) relate to UNKNOWN delay findings

  WLM4nn(series) relate to swap-in and target MPL findings

  WLM6nn(series) relate to Cross System Coupling Facility (XCF) findings

You may wish to read all of the rules in this appendix, just to see the type of problems that are encountered in different installations. However, it is not necessary to read all of the rules. It is necessary only to read the rules that apply to your installation. The rules that apply to your installation are identified by the report produced from the WLMCPE Module.

All references to *MVS Initialization and Tuning Guides* or *MVS Initialization and Tuning References* apply to the following specific documents:

MVS/XA Initialization and Tuning Guide, GC28-1149-4

MVS/ESA SP3.1 Initialization and Tuning Guide, GC28-1828-2

MVS/ESA SP4.1 Initialization and Tuning Guide, GC28-1634
MVS/ESA SP4.1 Initialization and Tuning Reference, GC28-1635

MVS/ESA SP4.2 Initialization and Tuning Guide, GC28-1634-3
MVS/ESA SP4.2 Initialization and Tuning Reference, GC28-1635-3

MVS/ESA SP4.3 Initialization and Tuning Guide, GC28-1634-4
MVS/ESA SP4.3 Initialization and Tuning Reference, GC28-1635-4

IBM released a new version of the *Initialization and Tuning Guide* and *Initialization and Tuning Reference* for SP4.3 in January 1994. The following documents are used for references updated after January 1994.

MVS/ESA SP4.3 Initialization and Tuning Guide, GC28-1634-5
MVS/ESA SP4.3 Initialization and Tuning Reference, GC28-1635-5

Beginning with MVS/ESA SP5.1, the references to IBM documents **apply to IBM BookManager documents**. This change was made because all CPExpert users installing MVS/ESA SP5.1 also use IBM BookManager to access soft-copy IBM documents rather than acquiring hard-copy IBM documents.

- The IBM BookManager documents are contained in IBM CDROM LK2T-5114 or in IBM CDROM SK2T-0710 (with appropriate quarterly updates).
- With OS/390, the IBM BookManager documents are contained in IBM CDROM SK2T-6700.
- With z/OS, the IBM BookManager documents are contained in IBM CDROM SK3T-4269.

If any user does not have access to IBM BookManager documents, please call Computer Management Sciences. We will be happy to provide references to hardcopy manuals.

Beginning with CICS/Transaction Server for z/OS, CICS documentation is contained in the CICS Information Center (InfoCenter). IBM provides the following description of the documentation available with CICS/Transaction Server for z/OS:

"For CICS Transaction Server V2.1 (announced March 2001), there has been a move away from printed books as the default deliverable to a new online concept. The primary source of user information for this release is a new CICS Information Center with a graphical user interface, delivered with the product on a CD-ROM. This HTML-based Information Center runs inside a Web browser, and provides a number of alternative means of accessing the information within it.

The objective of the Information Center is to make it easy for users to retrieve the information they need to perform specific CICS tasks, or to find relevant background or reference information on demand. At the heart of the Information Center is an HTML representation of the total CICS library (unlicensed books) Within the graphical user interface, the key documentation can be accessed via three main classes: tasks, concepts, and reference, each separately selectable. On selecting a class, the categories for that class are displayed in the navigation panel. Each of these can be expanded into a hierarchical navigation tree of topics in turn point to the detailed information.

The Information Center also includes a powerful search capability based on IBM's NetQuestion technology. Search results can be saved for future reference. In addition to the new methods of accessing the CICS documentation, the Information Center provides the more traditional alternative of a complete library listing of

the books, which can be viewed in both HTML and PDF formats. The latter also provides the capability to print either the whole book or some of the pages in hardcopy a printer, using Adobe Acrobat.

For this new release of CICS, the main focus of the documentation is the implementation of EJB technology in the CICS environment. A new piece of documentation entitled "Java Applications in CICS" is the cornerstone of this information, and has been designed to make use of the new capabilities of the Information Center."

CPExpert references for CICS/Transaction Server for z/OS are specific to the CICS Information Center.

List of Rules
Volume 1

RULE	DESCRIPTION
WLM001	The service class definition may not match workload
WLM002	Conflict exists between service class and report class
WLM003	The service policy was changed
WLM004	CPExpert believes too many service policy changes occurred
WLM005	The velocity goal may be too high for batch service class
WLM006	The response time goal is too large
WLM007	MSO service definition coefficient may be too large
WLM008	DUR value may be too large for TSO Period 1
WLM009	Minimum CPU service specified for Resource Group
WLM010	Velocity goals have values which are too similar
WLM011	The service definition does not describe all workloads
WLM012	A server workload defaulted to the SYSSTC service class
WLM013	Response goal was specified for a server service class
WLM014	Response goal specified for "hot batch" workload
WLM015	Execution velocity goal specified for TSO Period 1 or Period 2
WLM016	Low execution velocity goal specified for server service class
WLM017	Server and subsystem transactions in same service class
WLM018	Multiple periods specified for server service class

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 1

RULE	DESCRIPTION
WLM019	Multiple periods specified for subsystem transaction service class
WLM020	Subsystem transactions in same service class as address space
WLM021	Subsystem transactions service class assigned to resource group
WLM022	Execution velocity goal specified for subsystem transaction service class
WLM023	Too many service class periods may have been specified
WLM024	More than three periods were specified for a service class
WLM025	The service class period may be unnecessary
WLM026	Highest importance service class period had few samples
WLM027	Service class periods have same velocity goal and importance
WLM030	Report class period is heterogeneous
WLM031	Dynamic alias management was active but I/O priority management was not selected.
WLM032	Server was assigned CPU protection, but most work was done in support of lower importance work

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 1

RULE	DESCRIPTION
WLM050	The number of available page slots is low
WLM051	The number of local page data sets may be inadequate
WLM052	The number of allocated page slots may be insufficient
WLM053	The number of allocated page slots may be insufficient
WLM054	The number of allocated page slots may be insufficient
WLM055	Local page data sets are on same volume as swap data sets
WLM056	Local page data sets share volume with COMMON or PLPA
WLM057	Multiple local page data sets are on the same volume
WLM058	Local page response is significantly worse than average
WLM059	Insufficient local page data sets are defined for migration
WLM060	PLPA and COMMON page data sets may be combined
WLM061	Swap data sets are defined
WLM070	Terminal Output Wait swaps occur too often
WLM071	Detected Wait swaps occur too often
WLM080	JES-managed and WLM-managed job classes conflict
WLM081	WLM-managed job class assigned to multiple service classes
WLM082	Job might not be suitable for WLM-managed initiators
WLM090	SMF Type 30 interval recording not turned on

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM101	Service class did not achieve average response goal
WLM102	Service class did not achieve percentile response goal
WLM103	Service class did not achieve velocity goal
WLM104	Served service class did not achieve average response goal
WLM105	Served service class did not achieve percentile response goal
WLM106	Response time distribution for service class
WLM107	Response time distribution for service class
WLM108	Response time distribution for served service class
WLM109	Response time distribution for served service class
WLM110	BTE Phase samples count was larger than calculated samples
WLM111	BTE Phase Idle sample count is large
WLM112	BTE Phase had large (Ready plus Active) sample count
WLM113	BTE sample count was significantly less than calculated samples
WLM114	BTE phase had large ready samples
WLM115	Service class did not have begin_to_end phase samples
WLM116	Execution Phase samples did not exist in SMF data
WLM117	Transaction service class wait states
WLM119	Work manager data was not collected for service class

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM120	Significant transaction time was in Active state
WLM121	Significant transaction time was in Ready state
WLM122	Significant transaction time was in Idle state
WLM123	Significant transaction time was Waiting for Lock
WLM124	Significant transaction time was Waiting for I/O request
WLM125	Significant transaction time was Waiting for Conversation
WLM126	Significant transaction time was Waiting, Distributed
WLM127	Significant transaction time was Waiting, Local Session
WLM128	Significant transaction time was Waiting, Sysplex Session
WLM129	Significant transaction time was Waiting, Network Session
WLM130	Significant transaction time was Waiting for Timer
WLM131	Significant transaction time was Waiting, Another Product
WLM132	Significant transaction time was Waiting, Miscellaneous
WLM135	IMS activity processing transactions in service class
WLM136	DB2 activity processing transactions in service class
WLM140	Sysplex performance index was significantly less than local
WLM150	Server service class delays (single transaction service class)
WLM151	Server service class delays (multiple transaction service classes)
WLM152	Server served multiple transaction service classes

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM153	Server served multiple transaction service classes
WLM170	Address spaces were idle a significant percent of time
WLM171	Execution velocity was based on a small sample set
WLM172	Server was idle a significant percent of time
WLM173	The response performance goal may be too large

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM200	Average CPU use per transaction is higher than goal
WLM201	Goal may be unrealistic - average CPU use is high
WLM202	Average CPU use was a major cause of transaction delay
WLM210	Average server CPU use per transaction is higher than goal
WLM211	Goal may be unrealistic - average server CPU use is high
WLM212	Average CPU use was a major cause of transaction delay
WLM220	Service class was delayed because of resource capping
WLM221	Service Class was capped for discretionary goal management
WLM222	Service class was Active, but server was CPU capped
WLM250	Service class waited for access to CPU
WLM251	Dispatcher reduced preemption might have caused CPU delay
WLM252	CPU access might be denied because of Resource Group minimum
WLM255	Service class was active but server was denied CPU
WLM256	Service class was active and server was not denied CPU

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM340	Batch jobs may be delayed waiting for an initiator
WLM341	Service class may be waiting for initiator/scheduler
WLM350	I/O activity may have caused significant delays
WLM351	I/O activity may have caused significant delays
WLM352	I/O activity may have caused significant delays to server
WLM353	I/O activity may have caused significant delays to server
WLM355	Device DISConnect time was a major cause of DASD delays
WLM356	Device PEND time was a major cause of DASD delays
WLM357	Device CONNect time was a major cause of DASD delays
WLM358	Device IOS queuing time was a major cause of DASD delays
WLM359	I/O activity probably did not cause major delays
WLM360	Service class did not reference DASD

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM361	Non-paging DASD I/O activity caused significant delays
WLM362	Non-paging DASD I/O activity caused significant delays
WLM363	Non-paging DASD wait time was a major cause of DASD delays
WLM364	non-paging DASD CONNect time was a major cause of delays
WLM365	Non-paging DASD DISConnect time was a major cause of delays
WLM366	Non-paging DASD IOSQ time was a major cause of DASD delay
WLM370	Non-DASD I/O activtity or delay was significant
WLM371	Non-paging DASD I/O activity caused significant delays
WLM385	SMF Type 30 (Interval) data was not available for service class
WLM390	UNKNOWN delay was not accounted for by above analysis

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM400	Page-in from auxiliary storage was major performance problem
WLM410	Some higher importance service class has storage protection
WLM420	Some equal importance service class has storage protection
WLM450	Swap-in delay was major performance problem
WLM480	Target multiprogramming level delay was major performance problem

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM601	XCF transport class may need to be split
WLM602	XCF message buffer length may be too small
WLM603	XCF message buffer length may be too large
WLM604	XCF outbound message buffer space may be too small
WLM605	XCF inbound message buffer space may be too small
WLM606	XCF local message buffer space may be too small
WLM607	Insufficient outbound paths were defined
WLM608	Transport class did not have a signalling path assigned
WLM620	Message buffer space may be too small for inbound path
WLM621	Message buffer space may be too small for inbound path
WLM622	The number of outbound paths may need to be increased
WLM623	The number of outbound paths may need to be increased
WLM630	A hardware problem may exist
WLM632	An inbound path was non-operational
WLM633	An outbound path was non-operational

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM651	Lock contention was high
WLM652	False lock contention was high
WLM660	Service time was high for synchronous requests
WLM661	Service time was high for asynchronous requests
WLM662	Subchannel contention was high for synchronous requests
WLM665	Too many synchronous requests were changed to asynchronous

©Copyright 1994, Computer Management Sciences, Inc.

List of Rules (Continued)
Volume 2

RULE	DESCRIPTION
WLM701	Log stream coupling facility structure was full
WLM702	Log stream staging data set was full
WLM703	Log stream structure offloads occurred: 90% full
WLM704	Interim storage was not efficiently used for log stream
WLM705	Local storage buffers not efficiently used, DASD-only log stream
WLM706	DASD staging data set high threshold was reached
WLM707	Frequent log stream DASD-shifts occurred
WLM708	Log stream caused structure to reach high threshold
WLM709	Log stream consumed most of structure resources

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM101: Service Class did not achieve average response goal

Finding: CPExpert has detected that a service class period did not achieve the average response goal which was specified in the Service Policy in effect. This finding applies to performance goals which specify **average response time** as the performance goal.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The impact depends upon the importance of the service class which missed its performance goal, and on how seriously the goal was missed.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: The System Resources Manager (SRM) accounts for each transaction executing in the system. When the transaction ends, the SRM counts the transaction and determines the transaction's response time[1]. The SRM sums the response times for transactions ending in a service class period as each transaction ends.

The Workload Manager periodically[2] divides the sum of response times by the number of ending transactions. The result is the average response time of all transactions ending in the service class period during the previous interval.

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class[3]. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

For service classes which have an **average response time goal**, the Workload Manager determines whether the average response time achieved by transactions ending in the service class period is greater than the performance goal. If the average response time is greater than the

[1]This response time applies only to the time the transaction was in the system; it does not apply to response time delays experienced in the network.

[2]The Workload Manager computes the average transaction response time every 10 seconds, during the "policy adjustment" interval.

[3]Please see Section 4 for a more detailed description of this process.

©Copyright 1994, Computer Management Sciences, Inc.

performance goal, the system is not meeting performance goals for the service class period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

At a different period (typically every 15 minutes), the SRM provides RMF with measurement data, including the elapsed and execution times of transactions ending in each service class period, and the number of transactions ending in each service class period. This information is collected by RMF and written to the SMF data set as Type 72 records. The interval at which RMF collects data and writes records typically is referred to as the *RMF measurement interval*.

RMF does not include in Type 72 records the number of instances in which any service class period did not achieve its average response goal. RMF records the total elapsed time and the number of ending transactions.

For response goals, RMF also records in Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. See Rule WLM102 or Rule WLM105 for a discussion of percentile response performance goals.

The count of transactions completing in varying percentages of the performance goal is useful for analyzing performance of service classes which have a "percentile goal" specified for a service class. However, these counts are not useful in computing average response times.

CPExpert analyzes the SMF Type 72 records to determine whether service class periods met their performance goals during each RMF measurement interval. For service class periods which have an average response performance goal specified, CPExpert accomplishes this simply by dividing the **number** of transactions ending in the service class (R723CRCP) into the **elapsed time** of ending transactions (R723CTET). The result is the average transaction response time **over the entire RMF measurement interval**.

CPExpert compares the average transaction response time over the entire RMF measurement interval against the performance goal specified for the service class period. If the average transaction response time is greater than the performance goal, CPExpert can conclude that the service class period did not achieve its performance goal for the RMF measurement interval. **This conclusion reveals a persistent problem**.

Some transactions executing in the service class period may have missed their performance goals, and this situation is to be expected when an average response goal is specified to the Workload Manager. The average response goal simply applies to the *average* response time achieved, which implies that the response time of some transactions may be significantly *less* than the goal and others may be significantly *more* than the goal.

It is important to appreciate that the average response time goal may not be met during a number of Workload Manager policy adjustment intervals. This circumstance may not be detected when CPExpert analyzes RMF data as described above, as the averages are computed based on an entire RMF measurement interval. CPExpert will detect a **persistent** problem, but cannot detect **periodic** problems with average transaction response times being greater than the performance goal[4].

CPExpert produces Rule WLM101 when CPExpert detects that a service class period did not meet its average response goal for an entire RMF measurement interval. CPExpert reports the total transactions which ended during the interval, the average response achieved by the transactions, and the primary and secondary causes of response delay. Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the average transaction response time.

For example, suppose that a 100 millisecond average response time had been specified as the performance goal for a service class period serving interactive TSO transactions. CPExpert might detect that the average TSO response time was 350 milliseconds; the performance goal was missed by 250 milliseconds! CPExpert would report the number of transactions and their average response time.

CPExpert would analyze the causes of delay to TSO transactions and report the primary and secondary causes of delay. CPExpert might compute that the primary cause of delay to TSO transactions was that they were denied access to a processor for 35% of their active time, and that they were waiting for "unknown" causes[5] for another 30% of their active time.

---

[4]The Workload Manager does provide another category of service goal (the Percentile Goal) by which users can specify the percentage of transactions which should achieve their service goals. As mentioned earlier, the Percentile Goal is described in Rule WLM102 and Rule WLM105.

[5]Recall from Section 4 that the "unknown" cause is unknown as far as the Workload Manager is concerned. The Workload Manager identifies causes of delay only for those categories over which the SRM has control. Delays over which the SRM has no control are grouped together into an "unknown" category. These delays typically are I/O delay, ENQ delay, waiting for cross-memory services, etc.

CPExpert would report both these causes, and their respective percentages in Rule WLM101. CPExpert would continue analysis to assess which service classes might deprive TSO transactions from access to a processor and to assess the likely causes of "unknown" delays.

CPExpert analyzes the following possible delays to response time[6]:

- **CPU Using delay**
- **Denied CPU delay**
- **CPU Capping delay**
- **Swap-in delay**
- **MPL delay**
- **Page-in delay**
- **Non-paging DASD delay**
- **Non-DASD delay**
- **Queue delay**
- **Unknown delay**

The above causes of delay are analyzed by CPExpert in other rules.

For the purposes of identifying primary and secondary causes of response delay, CPExpert combines all auxiliary storage page-in delays into "page-in delay" to reflect the impact of auxiliary storage on response.

Additionally, CPExpert computes the average Performance Index for the service class during any measurement interval in which the performance goal was not achieved. The Performance Index is computed as the **actual response** divided by the performance **goal**.

The Performance Index gives an indication of how seriously the performance goal was missed: a Performance Index of less than 1 indicates that response was less than the performance goal; a Performance of greater than 1 indicates that response was worse than the performance goal.

[6]Please see Section 4 (Chapter 3.3) for a description of these delays.

The following example illustrates the output from Rule WLM101:

```
RULE WLM101:  SERVICE CLASS DID NOT ACHIEVE AVERAGE RESPONSE GOAL

   Service Class TSO (Period 1) did not achieve its response goal
   during the measurement intervals shown below.  The response goal was
   0.040 second average response, with an importance level of 2.  The
   percentages with the primary/secondary causes of delay are computed as
   a function of the average address space active time.

                              TOTAL    AVERAGE    PERF   PRIMARY,SECONDARY
   MEASUREMENT INTERVAL       TRANS   RESPONSE    INDX   CAUSES OF DELAY
   13:17-13:22,21JUN1994      5,750      0.055    1.39   DENIED CPU(51%),UNKNOWN(29%)
   13:22-13:27,21JUN1994      5,829      0.045    1.12   UNKNOWN(40%),DENIED CPU(36%)
```

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

Rule WLM102: Service Class did not achieve percentile response goal

Finding: CPExpert has detected that a service class period did not achieve the percentile response goal which was specified in the Service Policy in effect. This finding applies to performance goals which specify **percentile response time** as the performance goal.

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: Service classes can be defined which have a "percentile" response performance goal. A "percentile" response performance goal means that the performance goal is defined as "x%" of the transactions should complete within "y" time. For example, a typical percentile response goal is that **90% of the transactions should complete within 200 milliseconds**.

MVS accounts for each transaction executing in the system and determines the transaction's response time[1]. MVS maintains fourteen counters for each service class which has a response goal. The counters represent a response time distribution with respect to the response goal.

For response goals, RMF includes in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records[2].

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class[3]. This assessment is referred to as the "policy adjustment" interval. During the policy adjustment interval, the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

---

[1]This response time applies only to the time the transaction was in the system; it does not apply to response time delays experienced in the network.

[2]Please refer to Exhibit 4-11 in Section 4 for a description of the response time distributions.

[3]Please see Section 4 for a more detailed description of this process.

©Copyright 1994, Computer Management Sciences, Inc.

For service classes which have a **percentile response time goal**, the Workload Manager determines whether the specified percent of transactions are achieving the response time specified by the response goal for the service class. If more than the specified percent of transactions achieved a response greater than the specified response goal, the system is not meeting performance goals for the service class period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

CPExpert analyzes the SMF Type 72 records to determine whether service class periods met their performance goals during each RMF measurement interval. For service class periods which have a percentile response performance goal specified, the performance goal is specified as "x% of the transactions completing within y time." CPExpert simply sums the transaction count in the first six counters to determine the number of transactions ending within 100% or less of the response goal. This value is divided by the total number of transactions ending to yield the percent of transactions ending within 100% or less of the response goal. If the resulting percentage is less than the performance goal percentage, CPExpert can conclude that the performance goal was not met.

CPExpert produces Rule WLM102 when CPExpert detects that a service class period did not meet its percentile response goal for an entire RMF measurement interval. CPExpert reports the total transactions which ended during the interval, the number of transactions which met the response goal, the percentage of transactions which met the goal, and the primary and secondary causes of response delay.

Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the average transaction response time.

For example, suppose that an installation specified that 90% of the transactions should complete within 200 milliseconds for a service class period serving interactive TSO transactions. CPExpert might detect that only 80% of the transactions completed within 200 milliseconds, and the performance goal was not achieved. CPExpert would report the number of ending transactions, the number of transactions which met the 200 millisecond goal, and that only 80% of the transactions met the goal.

CPExpert would analyze the causes of delay to TSO transactions and report the primary and secondary causes of delay. CPExpert might compute that the primary cause of delay to TSO transactions was that they were denied access to a processor for 35% of their active time, and that

they were waiting for "unknown" causes[4] for another 30% of their active time. CPExpert would report both these causes, and their respective percentages in Rule WLM102. CPExpert would continue analysis to assess which service classes might deprive TSO transactions from access to a processor and to assess the likely causes of "unknown" delays.

CPExpert analyzes the following possible delays to response time[5]:

- **CPU Using delay**
- **Denied CPU delay**
- **CPU Capping delay**
- **Swap-in delay**
- **MPL delay**
- **Page-in delay**
- **Non-paging DASD delay**
- **Non-DASD delay**
- **Queue delay**
- **Unknown delay**

For the purposes of identifying primary and secondary causes of response delay, CPExpert combines all auxiliary storage page-in delays into "page-in delay" to reflect the impact of auxiliary storage on response.

Additionally, CPExpert computes the average Performance Index for the service class during any measurement interval in which the performance goal was not achieved. The Performance Index is computed as the actual response divided by the performance goal, but is a more detailed algorithm than the algorithm described in Rule WLM101[6].

The Performance Index gives an indication of how seriously the performance goal was missed: a Performance Index of less than 1

[4]Recall from Section 4 that the "unknown" cause is unknown as far as the System Resources Manager is concerned. The SRM identifies causes of delay only for those categories over which it has control. Delays over which the SRM has no control are grouped together into an "unknown" category. These delays typically are I/O delay, ENQ delay, waiting for cross-memory services, etc.

[5]Please see Section 4 (Chapter 3.3) for a description of these delays.

[6]Please refer to Section 4 for a description of how the Performance Index is computed for percentile performance goals.

indicates that response was less than the performance goal; a Performance of greater than 1 indicates that response was worse than the performance goal.

The following example illustrates the output from Rule WLM102:

```
RULE WLM102:  SERVICE CLASS DID NOT ACHIEVE PERCENTILE RESPONSE GOAL

   Service Class TSOUSERS (Period 1) did not achieve its response goal
   during the measurement intervals shown below.  The response goal was
   80.00 percent of the transactions completing within 0.500 seconds,
   with an importance level of 2.  The percentages with the primary/
   secondary causes of delay are computed as a function of the average
   address space active time.

                                  TRANS      %
                           TOTAL  MEETING  MEETING PERF   PRIMARY,SECONDARY
   MEASUREMENT INTERVAL    TRANS    GOAL     GOAL  INDX   CAUSES OF DELAY
   10:45-11:00,07DEC1994      63      37     58.7  2.00   UNKNOWN(53%),CPU USING(45%)
   11:15-11:29,07DEC1994      32      24     75.0  1.20   CPU USING(66%),UNKNOWN(32%)
   11:29-11:30,07DEC1994       1       0      0.0  4.00   UNKNOWN(100%)
   11:45-12:00,07DEC1994      14      10     71.4  4.00   UNKNOWN(73%),CPU USING(25%)
```

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

Rule WLM103: Service Class did not achieve execution velocity goal

Finding: CPExpert has detected that a service class period did not achieve the execution velocity goal which was specified in the Service Policy in effect.

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: Installations may specify an *execution velocity goal* for a service class period. An execution velocity is a measure of how fast work should run when the work is ready to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage[1]. The purpose of specifying an execution velocity goal is to allow installations to specify how important it is to have work processed, when the work has no time-related measure (that is, a response requirement is not associated with the work).

The execution velocity is computed based on samples collected at periodic sampling intervals[2] by the System Resources Manager (SRM). The SRM sampling code interrogates address space control blocks (TCBs, SRBs, OUCBs, and OUXBs) to determine the state of each address space assigned to a service class. Sampling counts associated with the service class are updated based upon the state[3] of the address spaces.

The sampling code records the sampling result into the following categories:

- **CPU using samples**. CPU using samples mean that the address space is using the CPU.

- **CPU delay samples**. CPU delay samples mean that the address space is ready to use the CPU but is being delayed. Two separate CPU delays are recorded:

---

[1]Processor storage is composed of *central storage* and *expanded storage*. The third category of storage is *auxiliary storage*.

[2]With MVS/ESA SP5.1, the sampling interval is 250 milliseconds. The state of each TCB or SRB associated with an address space is sampled every 250 milliseconds, beginning from address space initiation.

[3]Note that an address space can be in multiple states (for example, a CICS region might be using multiple processors concurrently, while some CICS tasks were also waiting on some function). Thus, the sample counts can total more than 100% of the sample intervals for any address space.

©Copyright 1994, Computer Management Sciences, Inc.

- **CPU delay**. CPU delay means that a TCB or SRB is waiting to be dispatched or a TCB is waiting for a local lock. CPExpert refers to this delay as "DENIED CPU" in various reports resulting from the analysis of Workload Manager constraints.

- **CPU Capping delay**. This delay to response time means that the maximum CPU service units had been consumed for the Resource Group to which the service class was assigned, and the Workload Manager had marked all address spaces associated with the Resource Group as non-dispatchable for some time-slice intervals.

  This delay does not necessarily mean that address spaces in the capped service class had consumed the CPU service units. The CPU service units could have been used by another service class if more than one service class had been assigned to the Resource Group.

- **Processor storage delay**. Processor storage delay samples means that an address space is ready to execute, but is delayed waiting for processor storage. Eight separate processor storage delays are recorded:

  - **Swap-in delay**. Swap-in delay means that the address space was delayed on swap-in (the swap-in had started, but had not completed).

  - **MPL delay**. MPL delay means that an address space was ready to be swapped in, but that the SRM had not initiated a swap-in because of target MPL constraints.

  - **Auxiliary page delay from private**. This page-in delay means that the address space experienced a page fault in the private area and the pages were coming from auxiliary storage.

  - **Auxiliary page delay from common**. This page-in delay means that the address space experienced a page fault in the Common area and the pages were coming from auxiliary storage.

  - **Auxiliary page delay from cross memory**. This page-in delay means that the address space experienced a page fault from cross memory and the pages were coming from auxiliary storage.

  - **Auxiliary page delay from VIO**. This page-in delay means that the address space experienced a page fault in VIO and the pages were coming from auxiliary storage.

  - **Auxiliary page delay from standard hiperspace**. This page-in delay means that the address space experienced a page fault from

standard hiperspace and the pages were coming from auxiliary storage.

- **Auxiliary page delay from ESO hiperspace**. IBM has defined this state to mean that the address space was experiencing page faults in ESO hiperspace and the pages were coming from auxiliary storage. Pages in ESO hiperspace are, by definition, resident only in expanded storage (ESO = Expanded Storage Only), and are never migrated to auxiliary storage. IBM offers the following explanation[4]:

  "The execution delay for ESO hiperspaces is a calculated value based on the assumption that if an application does a read for an ESO hiperspace page and that page is no longer available (has been cast out), the application will read the data from DASD somewhere. WLM/SRM takes the number of times a read failed in this way and multiplies it by the number of delay samples we expect a read of a page from DASD to represent and report the product as the execution delay samples for ESO hiperspace. This obviously is not a perfect solution, but we needed some way to get an estimate of how much delay is caused to an address space by not having enough expanded for an ESO hiperspace. Such an estimated is needed to properly manage the amount of expanded owned by the address space to the address space's goal."

Notice that only certain delay categories are included: only delays for processor or for processor storage are included in the "delay" category. These delays are under control of the SRM. Delays not under control of the SRM are not included in CPU or processor storage delays, but are included in an "unknown" delay category. **Unknown delay is not included in the execution velocity computation**.

For example, delay waiting for I/O completion is not under control of the SRM. Consequently, the Workload Manager does not include waiting for I/O completion in when it computes execution velocity. Rather, waiting for I/O completion is included in an "unknown" category when the SRM takes its samples. The "unknown" delay means that the SRM was unable to identify the cause of delay. In practice, this means that the delay was something over which the SRM had no control (e.g., I/O operations, ENQ delay, etc.).

The Workload Manager computes the execution velocity of a service class by applying the following algorithm:

---

[4]IBM TALKLink RMF FORUM appended at 15:39:18 on 95/05/29 GMT (by YOCOM at KGNVMC)
Subject: Workload Activity Report

$$\frac{using\ samples}{using\ samples + delay\ samples} * 100$$

where:

using samples include:

- The number of samples of work using the processor (CPU Using).
- The number of calculated samples of work using non-paging DASD I/O resources (DASD connect state or DASD disconnect state). I/O using samples are included only if the installation has elected to include WLM-managed I/O.

delay samples include:

- The number of samples of work delayed for the processor (Denied CPU Delay or CPU Capping delay).
- The number of samples of work delayed for processor storage. Delay for processor storage includes:
    - Paging delay
    - Swap-in delay
    - Swapped out for multiprogramming (MPL) reasons
    - Server address space creation delay
    - Initiation delays for batch jobs in WLM-managed job classes
- The number of calculated samples of work delayed for non-paging DASD I/O resources (DASD IOS queue delay, DASD subchannel pending delay, or DASD control unit queue delay). I/O delay samples are included only if the installation has elected to include WLM-managed I/O.

The result from the algorithm is multiplied by 100, to yield an execution velocity ranging from 0 (when the address space did not use the CPU) to 100 (when the address space was not delayed for any reason controlled by the SRM).

It is important to keep in mind that execution velocity applies **only to times when an address space is using a CPU or ready to use a CPU**. It does

not include times when an address space is idle, waiting for I/O, enqueued for a resource, etc.

The Workload Manager periodically[5] computes the execution velocity of all address spaces which have an execution velocity goal.

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

The actual comparison process is accomplished by computing a *Performance Index* for each service class[6]. For execution velocity goals, the performance index is computed by dividing the goal by the achieved velocity. If achieved velocity is greater than the goal, the performance index will be less than one. If achieved velocity is less than the goal, the performance index will be greater than one.

- For example, suppose that an execution goal of 30% had been specified. Further suppose that the execution velocity achieved was 50%. Dividing the goal by the achieved would yield a performance index of 0.6 (30%/50%=0.6).

- However, suppose that the execution velocity achieved was only 15%. Dividing the goal by the achieved would yield a performance index of 2.0 (30%/15%=2.0).

As can be seen by the above discussion, a performance index less than one implies that a performance goal **has** been met, while a performance index greater than one implies that a goal has **not** been. Thus, the performance index can be used to compare the performance of service classes, regardless of the type of performance goal specified for the service class[7].

For service classes which have an execution velocity goal, the Workload Manager determines whether the execution velocity is less than the performance goal. If the execution velocity is less than the performance goal, the system is not meeting performance goals for the service class

[5]The Workload Manager computes the execution velocity every 10 seconds, during the "policy evaluation" interval.

[6]Please see Section 4 for a discussion of how the Performance Index is computed and used.

[7]A discretionary goal has an implied performance index of 81%, which means that service classes with discretionary goals will always be considered as achieving their service goal.

period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

At a different period (typically every 15 minutes), the SRM provides RMF with measurement data, including the CPU Using, CPU Delay, and Storage Delay samples for each service class period. This information is collected by RMF and written to the SMF data set as Type 72 records. The interval in which RMF collects data and writes records typically is referred to as the *RMF measurement interval*.

CPExpert analyzes the SMF Type 72 records to determine whether service class periods met their performance goals during each RMF measurement interval. For service class periods which have an execution velocity performance goal specified, CPExpert accomplishes this simply by dividing the CPU Using samples (R723CCUS) by the total Using and Delay samples (R723CCUS + R723CTOT). The result is the average execution velocity **over the entire RMF measurement interval**.

CPExpert compares the average execution velocity over the entire RMF measurement interval against the performance goal specified for the service class period. If the average execution velocity is less than the performance goal, CPExpert can conclude that the service class period did not achieve its performance goal for the RMF measurement interval. **This conclusion reveals a persistent problem**.

It is important to appreciate that the execution velocity goal may not be met during a number of Workload Manager policy adjustment intervals. This circumstance may not be detected when CPExpert analyzes RMF data as described above, since the average execution velocity is computed by CPExpert is based on an entire RMF measurement interval. CPExpert will detect a **persistent** problem, but cannot detect **periodic** problems with execution velocities being less than the performance goal.

CPExpert produces Rule WLM103 when CPExpert detects that a service class period did not meet its execution velocity goal for an entire RMF measurement interval. CPExpert reports the percent CPU Using samples, percent total waiting samples, the resulting execution velocity, and the primary and secondary causes of delay. Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the address space delay.

CPExpert analyzes the following possible delays to service classes with an execution velocity goal[8]:

[8]Please see Section 4 (Chapter 3.3) for a description of these delays.

- **Denied CPU delay**
- **CPU Capping delay**
- **Swap-in delay**
- **MPL delay**
- **Page-in delay**
- **I/O delay**
- **Queue delay (Batch job initiator delay, TSO LOGON delay, or APPC request queue delay)**

The above causes of delay are analyzed by CPExpert in other rules.

For the purposes of identifying primary and secondary causes of response delay, CPExpert combines all auxiliary storage page-in delays into "page-in delay" to reflect the impact of auxiliary storage on response.

Notice that "CPU Using" is not included in the delays analyzed by CPExpert, as "CPU Using" is the **objective** of an execution velocity goal. Additionally, "Unknown" delay is not included in the delays analyzed by CPExpert, as "Unknown" delay is not included in the computation of execution velocity.

Each of the above causes of delay are analyzed by CPExpert in other rules.

The following example illustrates the output from Rule WLM103:

```
RULE WLM103:  SERVICE CLASS DID NOT ACHIEVE VELOCITY GOAL

   Service Class BATCHHI (Period 1) did not achieve its velocity goal
   during the measurement intervals shown below.  The velocity goal was
   10% execution velocity, with an importance level of 3.  The percentages
   are computed as a function of the average address space active time.

                          % CPU  % TOTAL  EXEC   PERF   PRIMARY,SECONDARY
   MEASUREMENT INTERVAL   USING  WAITING  VELOC  INDX   CAUSES OF DELAY
   13:07-13:12,21JUN1994    5.7   74.5      7%   1.40   DENIED CPU(74%)
   13:12-13:17,21JUN1994    6.3   78.7      7%   1.34   DENIED CPU(79%)
   13:17-13:22,21JUN1994    5.0   65.9      7%   1.43   DENIED CPU(66%)
```

Notice that there is no "SECONDARY" cause of delay shown in the example output from Rule WLM103. CPExpert lists a SECONDARY cause

of delay only if the delay is greater than the WLMSIG guidance variable.

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

Rule WLM104: Subsystem (transaction) Service Class did not achieve average response goal

Finding: CPExpert has detected that a service class did not achieve the average response goal which was specified in the Service Policy in effect. This finding applies to performance goals which specify **average response time** as the performance goal. Additionally, this finding applies to service classes which are part of a subsystem (e.g., CICS transactions). This finding is made only if subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5).

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

Notice that the **transactions** comprising the CICSPERS Service Class must actually execute in a CICS region executing CICS at a level of at least CICS/ESA Version 4.1. The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space (e.g., the CICS region) must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class may be one of

several "served" service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" may be related procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class).

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICSPERS Service Class, as CICSPERS is simply a logical entity which describes transactions and CICSPERS is not an address space. Rather, the Workload Manager allocates resources to the "server" address space (the CICSRGN Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICSPERS Service Class, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" service classes which are served by the "server" service class. It does this by allocating resources to the "server" service class. **These additional resources may (or may not) be used to provide service to the service class missing its goal**[1].

Suppose there are multiple "served" service classes associated with a "server" service class. If some "served" service class is failing to achieve its goal, the Workload Manager may allocate additional resources to the "server" service class. These additional resources might allow some "served" service classes to significantly exceed their performance goal and these "served" service classes may not be particularly important.

In our example, suppose that the CICSRGN Service Class is serving two service classes (the CICSPERS Service Class we described and a CICSADMN Service Class). Suppose that CICSPERS is important but that CICSADMN Service Class is of lower importance. If the Workload Manager detects that CICSPERS is not meeting its performance goal, the Workload Manager may allocate more resources to the CICSRGN Service Class. The CICSRGN would use the additional resources to provide

[1]Please refer to Section 4 for a more complete illustration of the "server" and "served" concepts.

service to both CICSPERS and CICSADMN. Consequently, CICSADMN might significantly exceed its performance goal[2].

To summarize this discussion, performance goals are associated with "served" service classes while resources are allocated to "server" service classes. Performance (i.e., transaction response time) is recorded at the "served" service class level, while resource use is recorded at the "server" service class level.

Subsystem service classes can be defined which have an "average" response goal or a "percentile" response performance goal. An "average" response goal means that the performance goal is defined as transactions should complete within an average of "y" time. A "percentile" response performance goal means that the performance goal is defined as "x%" of the transactions should complete within "y" time. For example, a typical percentile response goal is that **90% of the transactions should complete within 200 milliseconds**.

This rule (Rule WLM104) deals with performance goals for subsystem service classes which have an **average** response goal. Rule WLM105 deals with performance goals for subsystem service classes which have a **percentile** response goal.

The System Resources Manager (SRM) accounts for each transaction executing in the system and determines the transaction's response time[3]. The SRM sums the response times for transactions ending in a service class as each transaction ends. The Workload Manager periodically[4] divides the sum of response times by the number of ending transactions. The result is the average response time of all transactions ending in the service class during the previous interval.

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

For service classes which have an **average response time goal**, the Workload Manager determines whether the average response time

[2]Indeed, there is no guarantee that the additional resources would help CICSPERS unless CICSPERS had been properly **defined to CICS** as a higher priority than CICSADMN.

[3]This response time applies only to the time the transaction was in the system; it does not apply to response time delays experienced in the network.

[4]The Workload Manager computes the average transaction response time every 10 seconds, during the "policy evaluation" interval.

achieved by transactions ending in the service class is greater than the performance goal. If the average response time is greater than the performance goal, the system is not meeting performance goals for the service class. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

At a different interval (typically every 15 minutes), the SRM provides RMF with measurement data, including the elapsed and active times of transactions ending in each service class, and the number of transactions ending in each service class. This information is collected by RMF and written to the SMF data set as Type 72 records. The interval in which RMF collects data and writes records typically is referred to as the *RMF measurement interval*.

RMF does not include in Type 72 records the number of instances in which any service class did not achieve its average response goal. RMF records to total elapsed time and active times and the number of ending transactions.

For response goals, RMF also records in Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. See Rule WLM102 or Rule WLM105 for a discussion of percentile response performance goals.

The count of transactions completing in varying percentages of the performance goal is useful for analyzing performance of service classes which have a "percentile goal" specified for a service class. However, these counts are not useful in computing average response times.

CPExpert analyzes the SMF Type 72 records to determine whether service class met their performance goals during each RMF measurement interval. For service class which have an average response performance goal specified, CPExpert accomplishes this simply by dividing the number of transactions ending in the service class (R723CRCP) into the elapsed time of ending transactions (R723CTET). The result is the average transaction response time **over the entire RMF measurement interval**.

CPExpert compares the average transaction response time over the entire RMF measurement interval against the performance goal specified for the service class. If the average transaction response time is greater than the performance goal, CPExpert can conclude that the service class did not achieve its performance goal for the RMF measurement interval. **This conclusion reveals a persistent problem**.

Some transactions executing in the service class may have missed their performance goals, and this situation is to be expected when an average response goal is specified to the Workload Manager. The average response goal simply applies to the *average* response time achieved, which implies that the response time of some transactions may be significantly *less* than the goal and others may be significantly *more* than the goal.

It is important to appreciate that the average response time goal may not be met during a number of Workload Manager policy adjustment intervals. This circumstance may not be detected when CPExpert analyzes RMF data as described above, as the averages are computed based on an entire RMF measurement interval. CPExpert will detect a **persistent** problem, but cannot detect **periodic** problems with average transaction response times being greater than the performance goal[5].

CPExpert produces Rule WLM104 when CPExpert detects that a service class did not meet its average response goal for an entire RMF measurement interval. CPExpert reports the total transactions which ended during the interval, and the average response achieved by the transactions. Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the average transaction response time.

For example, suppose that a 100 millisecond average response time had been specified as the performance goal for a service class period serving CICS transactions. CPExpert might detect that the average response time was 350 milliseconds for transactions in the CICS subsystem service class; the performance goal was missed by 250 milliseconds! CPExpert would report the number of transactions and their average response time.

CPExpert would analyze the causes of delay to CICS transactions and report the primary and secondary causes of delay, **if the information is available**. Some subsystems may not provide detailed information about causes of delay[6]. If this case, CPExpert simply lists "data not available" under the primary and secondary causes of delay column.

The subsystem work manager (e.g., CICS) normally reports the causes of delay to the Workload Manager, using the Workload Management Services macros[7].

---

[5]The Workload Manager does provide another category of service goal (the Percentile Goal) by which users can specify the percentage of transactions which should achieve their service goals. As mentioned earlier, the Percentile Goal is described in Rule WLM102 and Rule WLM105.

[6]Early releases of IMS Version 5 did not correctly report transaction delays.

[7]Please refer to Section 4 (Chapter 2.2) for a description of the interaction between subsystems and the Workload Manager.

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* state and the *execution phase*. IMS Version 5 reports only *execution phase*.

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[8]. This action normally is done in a CICS TOR region.

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS AOR region.

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

In our example of CICS transactions, the CICS/ESA subsystem work manager would report transaction delays in the following states for the "served" service class:

- **Active state**. The active state indicates that there was a program executing on behalf of the work request in the "served" service class, from the perspective of the work manager. In the case of a CICS region, this means that a CICS task has been dispatched by CICS to process the transaction.

  However, the active state **does not mean that the task is executing** from the perspective of MVS. It simply means that the task has been dispatched by CICS. Other address spaces with a higher system dispatching priority could preempt the task dispatched by CICS, and these other address spaces could be using the CPU. The situation in which the CICS application task is denied use of the CPU is unknown to CICS.

- **Ready state**. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks

[8]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

ready to process transactions in the "served" service class than were dispatched by CICS.

- **Idle state**. The idle state indicates that there were no work requests (e.g., CICS transactions) ready to run in the service class.

- **Waiting for lock**. The waiting for lock state indicates that some work request (e.g., a CICS task) was waiting for a lock.

- **Waiting for I/O**. The waiting for I/O state indicates that the work manager was waiting for some I/O request on behalf of the "served" service class. This state could be waiting on an actual I/O operation or waiting on some other function related to the I/O request.

- **Waiting for conversation**. The waiting for conversation state indicates that the work manager was waiting for a response in a conversation mode.

- **Waiting for distributed request**. The waiting for distributed request state indicates that some function or data must be routed prior to resumption of the work request.

- **Waiting for session to be established locally**. The waiting for session to be established locally means a wait for a session to be established on the current MVS image.

- **Waiting for session to be established in sysplex**. The waiting for session to be established in sysplex means a wait for a session to be established somewhere in the sysplex.

- **Waiting for session to be established in network**. The waiting for session to be established in network means a wait for a session to be established somewhere in the network.

- **Waiting for timer**. The waiting for timer means that a work request was waiting for expiration of a timer.

- **Waiting for another product**. The waiting for another product means that a work request was waiting for another product to provide some service.

- **Waiting for a new latch**. The waiting for a new latch means that a work request was waiting for a new latch. A latch is a short-duration lock.

- **Waiting for unidentified resource**. The waiting for unidentified resource means that the work request was waiting, but that the work manager could not identify the cause of the wait.

The above causes of delay are analyzed by CPExpert in other rules.

The delays are recorded by RMF from two perspectives: (1) the *begin_to_end phase* of work requests in the service class and (2) the *execution phase* of work requests in the service class. CPExpert can analyze delays to transactions from both perspectives[9].

For SMF Type 72 records related to "server" service class (e.g., a CICS region), RMF records information identifying the service classes served by the server service class. This information is in the "Service Class Served Data Section" of the TYPE 72 records. If CPExpert discovers that a "served" service class did not achieve its performance goal, CPExpert identifies the "server" service classes which serve the service class not achieving its performance goal.

The following example illustrates the output from Rule WLM104:

```
RULE WLM104:  SERVICE CLASS DID NOT ACHIEVE AVERAGE RESPONSE GOAL

   Service Class CICUSRTX did not achieve its response goal during the
   measurement intervals shown below.  The response goal was 0.090 second
   average response, with an importance level of 2.  CICUSRTX was defined
   as a "served" Service Class (e.g., IMS or CICS transactions).  The
   below causes of delay (if available) were based upon EXECUTION PHASE
   samples.  CICUSRTX was served by CICSRGN.

                              TOTAL    AVERAGE     PRIMARY,SECONDARY
   MEASUREMENT INTERVAL       TRANS   RESPONSE     CAUSES OF DELAY
   13:07-13:12,21JUN1994     14,307      0.120     WAIT I/O(76%),READY(18%)
   13:17-13:22,21JUN1994     14,314      0.181     WAIT I/O(62%),READY(32%)
   13:22-13:27,21JUN1994     14,287      0.090     WAIT I/O(81%),READY(12%)
```

Recall that resources are allocated to "server" service classes, and these "server" service have information relating to resources used and relating to possible delays from a system view. After analyzing the information described above related to the "served" service class missing its performance goal, CPExpert analyzes the "server" service class to identify causes of delay from a system view.

In the example of Rule WLM104, CPExpert detected that the CICSUSRTX service class did not achieve its performance goal. After analyzing the

[9]A CPExpert guidance variable (the **PHASE** variable) in USOURCE(WLMGUIDE) controls which phase CPExpert initially analyzes. Please refer to Section 2 for a discussion of how the PHASE guidance variable may be used to direct CPExpert's analysis and why this guidance may be altered.

delays from the perspective of CICS, CPExpert will analyze the delays to the server (CICSRGN), from the perspective of the overall system.

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Finding: CPExpert has detected that a service class did not achieve the percentile response goal which was specified in the service policy in effect. This finding applies to performance goals which specify **percentile response time** as the performance goal. Additionally, this finding applies to service classes which are part of a subsystem (e.g., CICS transactions). This finding is made only if subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5).

Impact: This finding can have a HIGH IMPACT on performance of your computer system.

Logic flow: This is a basic finding. There are no predecessor rules.

Discussion: If subsystems are installed which support Workload Manager reporting (e.g., CICS/ESA Version 4.1 or IMS/ESA Version 5), installations can define service classes which describe particular transaction types and specify performance goals for the transactions in the service class. All transactions entering the system which fall into the workload category described by the service class are associated with the service class.

For example, an installation may wish to group all CICS transactions relating to personnel matters into a CICSPERS Service Class. The installation would define classification rules to the Workload Manager so all transactions relating to personnel matters would be placed into the CICSPERS Service Class. The installation would specify a performance goal for the CICSPERS Service Class, and an importance level for the goal.

Notice that the **transactions** comprising the CICSPERS Service Class must actually execute in a CICS region executing CICS at a level of at least CICS/ESA Version 4.1. The CICS region would report transaction performance information to the Workload Manager, and the Workload Manager would attempt to manage system resources to meet the performance goal specified for the CICSPERS Service Class.

The controlling address space must be in its own service class. In our example, suppose that the CICS region is placed into the CICSRGN Service Class. The CICSRGN Service Class would be considered a "server" and the CICSPERS Service Class may be one of several "served"

service classes controlled by the CICSRGN Service Class (other CICS service classes "served" by the CICSRGN "server" may be related procurement, administration, miscellaneous, etc.).

The CICSRGN will have its own performance goals and importance. However, these performance goals and importance are used by the Workload Manager **only at address space start-up** time. After the CICS region has started, its performance goals and importance are ignored by the Workload Manager. The Workload Manager will allocate resources based upon the performance goals and importance of the "served" service classes (in our example, the allocation will be based upon the performance of the CICSPERS transactions, and other "served" service classes served by the CICSRGN Service Class).

It is important to appreciate that the Workload Manager **does not** allocate resources to the CICSPERS Service Class, as CICSPERS is simply a logical entity which describes transactions and CICSPERS is not an address space. Rather, the Workload Manager allocates resources to the "server" address space (the CICSRGN Service Class). Similarly, the Workload Manager does not measure resources consumed by the CICSPERS Service Class, as CICS/ESA Version 4.1 does not report this information to the Workload Manager.

One implication of the structure of the "server" and "served" service classes is that the Workload Manager will attempt to meet the performance goals of all "served" service classes which are served by the "server" service class. It does this by allocating resources to the "server" service class. **These additional resources may (or may not) be used to provide service to the service class missing its goal**[1].

Suppose there are multiple "served" service classes associated with a "server" service class. If some "served" service class is failing to achieve its goal, the Workload Manager may allocate additional resources to the "server" service class. These additional resources might allow some "served" service classes to significantly exceed their performance goal and these "served" service classes may not be particularly important.

In our example, suppose that the CICSRGN Service Class is serving two service classes (the CICSPERS Service Class we described and a CICSADMN Service Class). Suppose that CICSPERS is important but that CICSADMN Service Class is of lower importance. If the Workload Manager detects that CICSPERS is not meeting its performance goal, the Workload Manager may allocate more resources to the CICSRGN Service Class. The CICSRGN would use the additional resources to provide

[1]Please refer to Section 4 for a more complete illustration of the "server" and "served" concepts.

service to both CICSPERS and CICSADMN. Consequently, CICSADMN might significantly exceed its performance goal. Indeed, there is no guarantee that the additional resources would help CICSPERS unless CICSPERS had been properly **defined to CICS** as a higher priority than CICSADMN.

To summarize this discussion, performance goals are associated with "served" service classes while resources are allocated to "server" service classes. Performance (i.e., transaction response time) is recorded at the "served" service class level, while resource use is recorded at the "server" service class level.

Service classes can be defined which have a "percentile" response performance goal. A "percentile" response performance goal means that the performance goal is defined as "x%" of the transactions should complete within "y" time. For example, a typical percentile response goal is that **90% of the transactions should complete within 200 milliseconds**.

This rule (Rule WLM105) deals with performance goals which have been specified as a **percentile response goal** (e.g., "x%" of the transactions should complete within "y" time). Rule WLM104 deals with performance goals for subsystem service classes which have an **average** response goal.

MVS accounts for each transaction executing in the system and determines the transaction's response time[2]. MVS maintains fourteen counters for each service class which has a response goal. The counters represent a response time distribution with respect to the response goal.

For response goals, RMF includes in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records[3].

The Workload Manager periodically assesses the performance of each service class, comparing the performance achieved by the service class against the performance goals specified for the service class. This assessment is referred to as the "policy adjustment" interval, in that the Workload Manager decides whether to adjust resource policies based on whether service classes are meeting performance goals.

---

[2]This response time applies only to the time the transaction was in the system; it does not apply to response time delays experienced in the network.

[3]Please refer to Exhibit 4-11 in Section 4 for a description of the response time distributions.

For service classes which have a **percentile response time goal**, the Workload Manager determines whether the specified percent of transactions are achieving the response time specified by the response goal for the service class. If more than the specified percent of transactions achieved a response greater than the specified response goal, the system is not meeting performance goals for the service class period. If the importance of the service class is sufficiently high, the Workload Manager may re-allocate system resources in an attempt to meet performance goals.

CPExpert analyzes the SMF Type 72 records to determine whether service class periods met their performance goals during each RMF measurement interval. For service class periods which have a percentile response performance goal specified, the performance goal is specified as "x% of the transactions completing within y time." CPExpert simply sums the transaction count in the first six counters to determine the number of transactions ending within 100% or less of the response goal. This value is divided by the total number of transactions ending to yield the percent of transactions ending within 100% or less of the response goal. If the resulting percentage is less than the performance goal percentage, CPExpert can conclude that the performance goal was not met.

CPExpert produces Rule WLM105 when CPExpert detects that a service class period did not meet its percentile response goal for an entire RMF measurement interval. CPExpert reports the total transactions which ended during the interval, the number of transactions which met the response goal, the percentage of transactions which met the goal, and the primary and secondary causes of response delay. Additionally, CPExpert computes the contribution that the primary and secondary causes of delay made to the average transaction response time.

For example, suppose that an installation specified that 90% of the transactions should complete within 100 milliseconds for a service class period serving CICS transactions. CPExpert might detect that only 80% of the transactions completed within 100 milliseconds, and the performance goal was not achieved. CPExpert would report the number of ending transactions, the number of transactions which met the 100 millisecond goal, and that only 80% of the transactions met the goal.

CPExpert would analyze the causes of delay to CICS transactions and report the primary and secondary causes of delay, **if the information is available**. Some subsystems may not provide detailed information about causes of delay[4]. If this case, CPExpert simply lists "data not available" under the primary and secondary causes of delay column.

[4]Early releases of IMS Version 5 did not correctly report transaction delays.

The subsystem work manager (e.g., CICS) normally reports the causes of delay to the Workload Manager, using the Workload Management Services macros[5].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*. IMS Version 5 reports only *execution phase* samples.

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[6]. This action normally is done in a CICS TOR region.

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS AOR region.

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

In our example of CICS transactions, the CICS/ESA subsystem work manager would report transaction delays in the following states for the "served" service class:

- **Active state**. The active state indicates that there was a program executing on behalf of the work request in the "served" service class, from the perspective of the work manager. In the case of a CICS region, this means that a CICS task has been dispatched by CICS to process the transaction.

  However, the active state **does not mean that the task is executing** from the perspective of MVS. It simply means that the task has been dispatched by CICS. Other address spaces with a higher system dispatching priority could preempt the task dispatched by CICS and these other address spaces could be using the CPU. The situation in which the CICS application task is denied use of the CPU is unknown to CICS[7].

---

[5]Please refer to Section 4 (Chapter 2.2) for a description of the interaction between subsystems and the Workload Manager.

[6]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

[7]The "denied CPU" state will be reported by the SRM in the CICSRGN service class, since the SRM samples control blocks for the CICS address space.

- **Ready state**. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to process transactions in the "served" service class than were dispatched by CICS.

- **Idle state**. The idle state indicates that there were no work requests (e.g., CICS transactions) ready to run in the service class.

- **Waiting for lock**. The waiting for lock state indicates that some work request (e.g., a CICS task) was waiting for a lock.

- **Waiting for I/O**. The waiting for I/O state indicates that the work manager was waiting for some I/O request on behalf of the "served" service class. This state could be waiting on an actual I/O operation or waiting on some other function related to the I/O request.

- **Waiting for conversation**. The waiting for conversation state indicates that the work manager was waiting for a response in a conversation mode.

- **Waiting for distributed request**. The waiting for distributed request state indicates that some function or data must be routed prior to resumption of the work request.

- **Waiting for session to be established locally**. The waiting for session to be established locally means a wait for a session to be established on the current MVS image.

- **Waiting for session to be established in sysplex**. The waiting for session to be established in sysplex means a wait for a session to be established somewhere in the sysplex.

- **Waiting for session to be established in network**. The waiting for session to be established in network means a wait for a session to be established somewhere in the network.

- **Waiting for timer**. The waiting for timer means that a work request was waiting for expiration of a timer.

- **Waiting for another product**. The waiting for another product means that a work request was waiting for another product to provide some service.

- **Waiting for a new latch**. The waiting for a new latch means that a work request was waiting for a new latch. A latch is a short-duration lock.

- **Waiting for unidentified resource**. The waiting for unidentified resource means that the work request was waiting, but that the work manager could not identify the cause of the wait.

The above causes of delay are analyzed by CPExpert in other rules.

Additionally, CPExpert could report that the “delay” was because the transaction was switched to a local MVS image, switched to another system in the sysplex, or switched to some system in the network.

- If the transaction was switched to a local MVS image, CPExpert can perform further analysis on the information for the current system.

- If the transaction was switched to another system in the sysplex, CPExpert will analyze other systems on which the service class appears. Information will be provided about delays to the service class on these other systems.

- If the transaction was switched to some system in the network, no information is available in the SMF data and no further analysis can be done.

The delays are recorded by RMF from two perspectives: (1) the *begin_to_end phase* of work requests in the service class and (2) the *execution phase* of work requests in the service class. CPExpert can analyze delays to transactions from both perspectives[8].

Additionally, some service classes might have *begin_to_end phase* data, but might **not** have *execution phase* data. In this case (and if the basic analysis is based on *execution phase* data), CPExpert will indicate “NO EXE PHASE DATA” in the PRIMARY,SECONDARY CAUSES OF DELAY, and will provide information about the *begin_to_end phase*. Rule WLM116 provides information for this situation.

For SMF Type 72 records related to "server" service class (e.g., a CICS region), RMF records information identifying the service classes served by the server service class. This information is in the "Service Class Served Data Section" of the TYPE 72 records. If CPExpert discovers that a "served" service class did not achieve its performance goal, CPExpert

[8]A CPExpert guidance variable (the **PHASE** variable) in USOURCE(WLMGUIDE) controls which phase CPExpert initially analyzes. Please refer to Section 2 for a discussion of how the PHASE guidance variable may be used to direct CPExpert's analysis and why this guidance may be altered.

identifies the "server" service classes which serve the service class not achieving its performance goal.

The following example illustrates the output from Rule WLM105:

```
RULE WLM105:  SERVICE CLASS DID NOT ACHIEVE PERCENTILE RESPONSE GOAL

   Service Class CICADMTX did not achieve its response goal during the
   measurement intervals shown below.  The response goal was 75.0 percent
   of the transactions completing within 0.090 seconds, with an importance
   level of 3.  CICADMTX was defined as a "served" Service Class (e.g.,
   IMS or CICS transactions).  The below causes of delay were based upon
   BEGIN_TO_END PHASE samples.  CICADMTX was served by CICSRGN.

                                   TRANS      %
                            TOTAL  MEETING  MEETING PERF  PRIMARY,SECONDARY
   MEASUREMENT INTERVAL     TRANS    GOAL     GOAL  INDX  CAUSES OF DELAY
   13:02-13:07,21JUN1994  14,326   9,463     66.1  4.00  WAIT I/O(65%),READY(22%)
   13:07-13:12,21JUN1994  14,307   8,709     60.9  4.00  WAIT I/O(52%),READY(35%)
   13:12-13:17,21JUN1994  14,357   9,216     64.2  4.00  WAIT I/O(65%),READY(25%)
   13:17-13:22,21JUN1994  14,314   8,669     60.6  4.00  WAIT I/O(40%),READY(51%)
   13:22-13:27,21JUN1994  14,287   9,172     64.2  4.00  WAIT I/O(63%),READY(32%)
   13:27-13:30,21JUN1994   8,612   5,639     65.5  4.00  WAIT I/O(65%),READY(29%)
```

Recall that resources are allocated to "server" service classes, and these "server" service have information relating to resources used and relating to possible delays from a system view. After analyzing the information described above related to the "served" service class missing its performance goal, CPExpert analyzes the "server" service class to identify causes of delay from a system view.

In the example of Rule WLM105, CPExpert detected that the CICSADMTX service class did not achieve its performance goal. After analyzing the delays from the perspective of CICS, CPExpert will analyze the delays to the server (CICSRGN), from the perspective of the overall system.

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules will be produced to provide more information.

Rule WLM106: Response time distribution for service class with average response performance goal

Finding: This rule provides information about the distribution of response times during those intervals when the identified service class missed its performance goal.

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided to allow you to assess the overall performance of service classes having an average response time performance goal.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM101: Service Class did not achieve average response goal

Discussion: For service classes with response goals, RMF includes in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. Section 4 describes the percentages recorded by RMF;

When CPExpert produces Rule WLM101, CPExpert automatically produces Rule WLM106 to provide a *summary* distribution of the response information. The purpose of Rule WLM106 is to allow you to assess whether the average response finding is meaningful, or whether there are some transactions which skew the averages.

The following example illustrates the output from Rule WLM106.

In the example, notice that 0.7% of the transactions had a response of over 400% of the 0.200 second goal. The data do not show the actual response time, but over 400% of the goal corresponds to at least 0.800 second response (0.200 second goal * 400% = 0.800). In this example, 0.7% of 137 transactions represents only 1 transaction. Thus, 1 transaction had an extremely long response, while most of the transactions experienced a response of less than 50% of the goal (or less than 0.100 seconds).

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM106:  RESPONSE TIME DISTRIBUTION FOR SERVICE CLASS

   Service Class TSOUSERS (Period 2) did not achieve its average response
   goal during the measurement intervals shown below.  The response goal
   was 0.200 second average response.  Average response can be misleading,
   since extremes can skew the average.  The below information shows the
   distribution of response times:

                                  --PERCENT COMPLETIONS RELATIVE TO GOAL--
                                         50-   90-  100-  110-  200-
                           TOTAL  <50%   90%  100%  110%  200%  400% >400%
   MEASUREMENT INTERVAL    TRANS  GOAL  GOAL  GOAL  GOAL  GOAL  GOAL  GOAL
   12:00-12:15,08NOV1994     137  98.5   0.0   0.7   0.0   0.0   0.0   0.7
```

Suggestion: If you find that some transactions skewed the findings, you may wish to consider the following alternatives:

- Since you specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

- You can simply ignore the findings which CPExpert made associated with this service class for the interval. You may decide that the transaction response is an anomaly and not take any further action. In the example shown above, only one transaction had a response significantly over the goal. It may be unnecessary to take action based on a small number of transactions exceeding the performance goal.

Rule WLM107: Response time distribution for service class with percentile response performance goal

Finding: This rule provides information about the distribution of response times during those intervals when the identified service class missed its performance goal.

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided to allow you to assess the overall performance of service classes having a percentile response time performance goal.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM102: Service Class did not achieve average response goal

Discussion: For service classes with response goals, RMF includes in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. Section 4 describes the percentages recorded by RMF.

When CPExpert produces Rule WLM102, CPExpert automatically produces Rule WLM107 to provide a summary distribution of the response information. The purpose of Rule WLM107 is to allow you to assess whether the response is meaningful, or whether there are some transactions which skew the finding.

The following example illustrates the output from Rule WLM107.

```
RULE WLM107:  RESPONSE TIME DISTRIBUTION FOR SERVICE CLASS

   Service Class TSOUSERS (Period 1) did not achieve its response goal
   during the measurement intervals shown below. The response goal was
   80.00 percent of the transactions completing within 0.500 seconds.
   The below information shows the distribution of response times:

                                  --PERCENT COMPLETIONS RELATIVE TO GOAL--
                                         50-   90-  100-  110-  200-
                            TOTAL  <50%   90%  100%  110%  200%  400% >400%
   MEASUREMENT INTERVAL     TRANS  GOAL  GOAL  GOAL  GOAL  GOAL  GOAL  GOAL
   10:45-11:00,07DEC1994       63  52.4   4.8   1.6   0.0   0.0  12.7  28.6
   11:15-11:29,07DEC1994       32  40.6  31.3   3.1   3.1   0.0   6.3  15.6
   11:29-11:30,07DEC1994        1   0.0   0.0   0.0   0.0   0.0   0.0 100.0
   11:45-12:00,07DEC1994       14  64.3   0.0   7.1   0.0   7.1  14.3   7.1
```

Suggestion: If you find that some transactions skewed the findings, you may wish to consider the following alternatives:

- In the example shown above, there seemed to be a bimodal distribution of response: many transactions experienced a response time of less than 50% of the goal while many transactions experienced a response time of greater than 200% of the goal.

  - The bimodal distribution may indicate that the service class contains transactions with dissimilar characteristics. In this case, perhaps you can use Workload Categorization to place the transactions into a different service class if you can identify the transactions.

    You may wish to specify a different importance and different performance goal for this new service class. Other findings by CPExpert may bolster this conclusion if (for example) CPExpert notes that the service class required a significant amount of CPU per average transaction (see Rule WLM200 for a discussion of this situation).

  - The bimodal distribution may indicate that there are system problems which causes the poor response of some transactions in the service class.

    Other service classes may interfere with the service class missing its performance goal. This situation would typically be identified by a subsequent finding by CPExpert that address spaces in the service class was "denied CPU" by other address spaces[1] (see Rule WLM255 for a discussion of this situation).

    Alternatively, CPExpert might identify DASD DISConnect time as a likely cause of delay (see Rule WLM355 for a discussion of this situation). DASD DISConnect time is primarily composed of seek time or missed RPS reconnect delay. Seek time or missed RPS reconnect delays often are caused by I/O activity by other address spaces referencing the I/O subsystem.

- You can simply ignore the findings which CPExpert made associated with this service class for the interval. You may decide that the poor transaction response is an anomaly and not take any further action.

[1]Address spaces in the service class could be "denied CPU" by address spaces in other service classes, by system functions, or by address spaces in the service class itself competing with each other.

Rule WLM108: Response time distribution for subsystem service class with average response performance goal

Finding: This rule provides information about the distribution of response times during those intervals when the identified service class missed its performance goal. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided to allow you to assess the overall performance of service classes having an average response time performance goal.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Discussion: For service classes with response goals, RMF records in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. Section 4 describes the percentages recorded by RMF.

When CPExpert produces Rule WLM104, CPExpert automatically produces Rule WLM108 to provide a *summary* distribution of the response information. The purpose of Rule WLM108 is to allow you to assess whether the average response finding is meaningful, or whether there are some transactions which skew the averages.

Suggestion: Please refer to the documentation for Rule WLM106 for additional discussion of the distribution of response times and suggestions for alternative actions based on the results.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM109: Response time distribution for subsystem service class with percentile response performance goal

Finding: This rule provides information about the distribution of response times during those intervals when the identified service class missed its performance goal. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided to allow you to assess the overall performance of service classes having an average response time performance goal.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: For service classes with response goals, RMF includes in SMF Type 72 records a count of transactions which completed in varying percentages of the response goal. These transaction counts are recorded by RMF as the "Response Time Distribution Count Table" contained in SMF Type 72(Subtype 3) records. Section 4 describes the percentages recorded by RMF.

When CPExpert produces Rule WLM105, CPExpert automatically produces Rule WLM109 to provide a *summary* distribution of the response information. The purpose of Rule WLM109 is to allow you to assess whether the average response finding is meaningful, or whether there are some transactions which skew the averages.

Suggestion: Please refer to the documentation for Rule WLM107 for additional discussion of the distribution of response times and suggestions for alternative actions based on the results.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM110: BTE Phase samples were larger than calculated samples

Finding: CPExpert has detected that the number of begin_to_end (BTE) phase samples recorded in the SMF Type 72 records were larger than the total number of samples which would be collected based upon the transaction elapsed time. This finding applies only to service classes representing CICS/ESA Version 4 transactions.

Impact: This finding means that long-running or never-ending transactions processed in the service class. The presence of these transactions can distort response time calculations, particularly with standard reports produced by RMF.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

CICS provides the System Resources Manager (SRM) with information

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

about the phase (begin_to_end phase or execution phase) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

The SRM also keeps a count of the number of samples which it takes of the begin_to_end phase and of the execution phase. The counts of various samples are recorded in the "Work Manager/Resource Manager State Section" of SMF Type 72 records.

The SRM also includes the elapsed time of transactions (R723CTET) and the count of transactions (R723CRCP) in the SMF Type 72 records. Based on the transaction elapsed time and transaction count, CPExpert can compute the approximate number of samples which the SRM **should** take of the begin_to_end phase of transactions. Comparing the results of this computation against the actual number of begin_to_end samples reveals valuable information about the nature of the transactions.

To illustrate the computation, suppose that a single transaction were to execute in a service class, and further suppose that the transaction elapsed time was 1 second. During this second of elapsed time, the SRM should take a sample every 250 milliseconds (4 samples per second), or

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

4 samples of the begin_to_end phase[4] of the transaction of the 1-second transaction. If two transactions with individual elapsed times of 1 second were to execute in the service class, the SRM should take 8 samples (1 second average elapsed time * 2 transactions * 4 samples per second = 8).

Thus, the computation of the number of samples which the SRM **should** take in any RMF measurement interval is simply the total elapsed time of transactions, times the sampling rate. The result from this computation should never be less than the number of samples which the SRM took of the begin_to_end phase, since the begin_to_end phase does not start until after the transaction has entered the system and has been classified to a service class, and the begin_to_end phase ends before the transaction is finally marked "ended" by the SRM.

Unfortunately, the result of the computation sometimes results in the number of begin_to_end phase samples being larger than the samples the SRM should take based on the elapsed time of transactions. This situation can occur when never-ending or long-running transactions execute in the service class.

The SRM updates the elapsed time of transactions only when the transactions end. Suppose that a never-ending transaction executed in the service class. The SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase). However, the SRM would never see the transaction complete and thus would not update the elapsed time of the transaction.

A similar situation occurs with long-running transactions. These transactions can span RMF measurement intervals; the SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase) in one RMF interval. The elapsed time of the transaction might not be recorded until a subsequent RMF interval.

These anomalies can cause response time calculations to be misleading, as discussed in Section 4. More importantly, the Workload Manager algorithms may be less effective if never-ending or long-running transactions are in the same service class as interactive transactions. This is because the Workload Manager's computation of response times may be distorted by the long-running transactions.

[4]For the moment, we can ignore the time required by the SRM to assign the transaction to a CICS region, the time for the CICS region to issue the IWMCLSFY macro, the time for the Workload Manager to classify the transaction to a service class, and the time for the CICS TOR to issue the IWWMINIT macro. These times normally are very small.

CPExpert can identify situations when the computed number of samples is significantly different from the expected number of samples. If the begin_to_end phase sample count is larger than the computed number of samples, CPExpert can confidently conclude that there were long-running or never-ending transactions executing in the service class. CPExpert produces Rule WLM110 when the number of begin_to_end samples is larger than the number of computed samples to advise you that long-running or never-ending transactions executed in the service class for which you have specified a response goal.

The following example illustrates the output from Rule WLM110:

```
RULE WLM110:  BTE PHASE SAMPLES WERE LARGER THAN CALCULATED SAMPLES

   CPExpert has detected that the BEGIN_TO_END PHASE samples recorded for
   the CICUSRTX Service Class were larger than the total samples which would
   be taken based on the transaction elapsed time and the sampling rate.
   This means that there were long-running transactions or never-ending
   transactions executing in the CICUSRTX Service Class.  Please refer to
   the WLM Component User Manual for a discussion of the implications of
   this finding.

                                BEGIN TO END       CALCULATED
   MEASUREMENT INTERVAL         PHASE SAMPLES        SAMPLES
   13:02-13:07,21JUN1994            4,733               103
   13:07-13:12,21JUN1994            6,426               145
   13:12-13:17,21JUN1994            4,844               108
   13:17-13:22,21JUN1994           10,041               218
   13:22-13:27,21JUN1994            4,906               108
```

Suggestion: CPExpert suggests that you identify the never-ending or long-running transactions and remove them for the service class identified by Rule WLM110. Since the CICS transactions are never-ending or long-running, it makes no sense to have the transactions in a service class with an interactive response goal.

IBM suggests the following guidance for CICS transactions:

- Do not mix CICS-supplied transactions with user transactions
- Do not mix routed with non-routed transactions
- Do not mix conversational with pseudo-conversational transactions
- Do not mix long-running and short-running transactions.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.3 Performance Guide
Section 2.5.1.7: Setting up service definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM111: BTE Phase IDLE sample count is large

Finding: CPExpert has detected that a large percent of the begin_to_end (BTE) phase samples were in IDLE state. This finding applies only to service classes representing CICS/ESA Version 4 transactions.

Impact: This finding means that conversational transactions were processed in the service class. The presence of these conversational transactions can distort response time calculations and corrupt the analysis performed by CPExpert.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

One of the states reported by CICS is the IDLE state. The idle state indicates that there were no work requests (e.g., CICS transactions) ready to run in the service class. When the IDLE state is reported for the begin_to_end phase, the IDLE state means that the CICS transaction is waiting on the results from a terminal (that is, a conversational transaction is waiting on a response from a terminal operator).

The service class being analyzed by CPExpert exceeded its performance goal (as reported by Rule WLM104 or Rule WLM105). However, the response for the transaction includes the time the terminal operator takes to formulate and enter a response. Unfortunately, this response time is included in the calculation of system response (the transaction is still active, but it is dependent upon a terminal operator response).

Terminal operator response time normally is unpredictable and the time can be quite lengthy, especially when compared with the normal system response time. The terminal operator response time should not be included in the calculation of a performance goal, since the Workload Manager cannot manage system resources to meet the performance goal of the service class when response time is a function of delays caused by a terminal operator.

CPExpert produces Rule WLM111 when the IDLE samples account for more than 25% of the number of begin_to_end samples AND when you have directed CPExpert to analyze response delays based on the

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

begin_to_end phase[4]. Since CPExpert is analyzing response delays based on begin_to_end phase samples, Rule WLM111 advises you that the analysis is significantly corrupted by the large number of IDLE samples.

The following example illustrates the output from Rule WLM111:

```
RULE WLM111:  BTE PHASE IDLE SAMPLE COUNT IS LARGE

  CPExpert has detected that the BEGIN_TO_END PHASE Idle samples recorded

  for the CICUSRTX Service Class is quite large.  This means that there were
  conversational transactions executing in the | Service Class, and these
  conversational transactions distort the response times.  Please refer to
  the WLM Component User Manual for a discussion of the implications of this
  finding.


                                BEGIN TO END          IDLE        %  IDLE
  MEASUREMENT INTERVAL          PHASE SAMPLES        SAMPLES      SAMPLES
   13:22-13:27,21JUN1994            4,906             2,302         47.9
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- Identify the conversational transactions and remove them for the service class identified by Rule WLM111. Since the CICS transactions are conversational, it makes no sense to have the transactions in a service class with an interactive response goal.

  IBM suggests the following guidance for CICS transactions:

  - Do not mix CICS-supplied transactions with user transactions
  - Do not mix routed with non-routed transactions.
  - Do not mix conversational with pseudo-conversational transactions
  - Do not mix long-running and short-running transactions.

- Change the guidance to CPExpert such that CPExpert analyzes delays in the execution phase of the transactions. This is done by specifying **%LET PHASE=EXECUTION;** in USOURCE(WLMGUIDE). With this specification, CPExpert will analyze delays in the execution phase and

[4]That is, you had specified **%LET PHASE=BEGIN_TO_END** in USOURCE(WLMGUIDE).

will mostly ignore the begin_to_end phase. The begin_to_end phase samples are relatively meaningless for this service class since such a large amount of response time was spent in IDLE state waiting on a conversation.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM112: BTE Phase had large number of Active plus Ready samples

Finding: CPExpert has detected that a large percent of the begin_to_end (BTE) phase samples were in the Active state or Ready state. This finding applies only to service classes representing CICS/ESA Version 4 transactions.

Impact: This finding means that non-routed transactions were processed in the service class. The presence of these transactions can distort response time calculations.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

Included in the state reported by CICS are the times the transaction is in an Active state and the times the transaction is in a Ready state.

- **Active state**. The active state indicates that there was a program executing on behalf of the work request in the "served" service class, from the perspective of the work manager. In the case of a CICS region, this means that a CICS task has been dispatched by CICS to process the transaction.

  However, the active state **does not mean that the task is executing** from the perspective of MVS. It simply means that the task has been dispatched by CICS. Other address spaces with a higher system dispatching priority could preempt the task dispatched by CICS and these other address spaces could be using the CPU. The situation in which the CICS application task is denied use of the CPU is unknown to CICS.

- **Ready state**. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to execute in the "served" service class than were dispatched by CICS.

CICS transactions typically enter the system via a CICS TOR. The transactions receive some initial processing in the TOR and are routed to

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

an AOR for actual application processing. CICS signals the beginning of the execution phase for the transaction when the transaction is received by the AOR.

Some transactions are not routed to an AOR, however. These transactions are completely processed in the TOR. Since the AOR signals the beginning of the execution phase, these transactions never enter the execution phase. Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

If non-routed transactions are processed in a service class with a response objective, the non-routed transactions can distort response time calculations.

The service class being analyzed by CPExpert exceeded its performance objective (as reported by Rule WLM104 or Rule WLM105). Further, CPExpert had been directed to analyze response time based on the execution phase[4].

CPExpert produces Rule WLM112 when the Active samples plus Ready samples account for more than 25% of the number of begin_to_end samples AND when you have directed CPExpert to analyze response delays based on the execution phase. CPExpert concludes that a large percentage of non-routed transactions are processed in the service class if more than 25% of the transaction samples occurred in the Active state and Ready state of the begin_to_end phase.

Since CPExpert is analyzing response delays based on execution phase samples, Rule WLM112 advises you that the analysis is significantly corrupted by the large number of non-routed transactions. Further, the Workload Manager's algorithms will be less effective if non-routed transactions are assigned to the same service class as routed transactions.

Suggestion: CPExpert suggests that you identify the non-routed transactions and remove them for the service class identified by Rule WLM112. Since the CICS transactions are non-routed, they should not be included in the same service class as routed transactions.

IBM suggests the following guidance for CICS transactions:

- Do not mix CICS-supplied transactions with user transactions

[4]That is, you had specified **%LET PHASE=EXECUTION** in USOURCE(WLMGUIDE).

- Do not mix routed with non-routed transactions.
- Do not mix conversational with pseudo-conversational transactions
- Do not mix long-running and short-running transactions.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM113: BTE Phase samples were significantly less than total calculated samples

Finding: CPExpert has detected that the number of begin_to_end (BTE) phase samples recorded in the SMF Type 72 records were less than the total number of samples which would be collected based upon the transaction elapsed time. This finding applies only to service classes representing transactions for CICS/ESA Version 4 (and subsequent versions).

Impact: This finding means that long-running or never-ending transactions processed in the service class. The presence of these transactions can distort response time calculations, particularly with standard reports produced by RMF.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: Beginning with CICS/ESA Version 4.1, CICS reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 or CICS/TS has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1, CICS/TS, or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

  Some CICS transactions may never enter the execution phase, as the transactions will be completely processed in the CICS TOR. These CICS transactions are termed "non-routed" transactions.

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

Consequently, the number of transactions completing the execution phase may be less than the total number of CICS transactions processed by the system.

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

The SRM also keeps a count of the number of samples which it takes of the begin_to_end phase and of the execution phase.

The counts of various samples are recorded in the "Work Manager/Resource Manager State Section" of SMF Type 72 records.

The SRM also includes the elapsed time of transactions (R723CTET) and the count of transactions (R723CRCP) in the SMF Type 72 records. Based on the transaction elapsed time and transaction count, CPExpert can compute the approximate number of samples which the SRM **should** take of the begin_to_end phase of transactions. Comparing the results of this computation against the actual number of begin_to_end samples reveals valuable information about the nature of the transactions.

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

To illustrate the computation, suppose that a single transaction were to execute in a service class, and further suppose that the transaction elapsed time was 1 second. During this second of elapsed time, the SRM should take a sample every 250 milliseconds (4 samples per second), or 4 samples of the begin_to_end phase[4] of the transaction of the 1-second transaction. If two transactions with individual elapsed times of 1 second were to execute in the service class, the SRM should take 8 samples (1 second average elapsed time * 2 transactions * 4 samples per second = 8).

Thus, the computation of the number of samples which the SRM **should** take in any RMF measurement interval is simply the total elapsed time of transactions, times the sampling rate. The result from this computation should never be less than the number of samples which the SRM took of the begin_to_end phase, since the begin_to_end phase does not start until after the transaction has entered the system and has been classified to a service class, and the begin_to_end phase ends before the transaction is finally marked "ended" by the SRM.

Unfortunately, the result of the computation sometimes results in the number of begin_to_end phase samples being larger than the samples the SRM should take based on the elapsed time of transactions. This situation can occur when never-ending or long-running transactions execute in the service class.

The SRM updates the elapsed time of transactions only when the transactions end. Suppose that a never-ending transaction executed in the service class. The SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase). However, the SRM would never see the transaction complete and thus would not update the elapsed time of the transaction.

A similar situation occurs with long-running transactions. These transactions can span RMF measurement intervals; the SRM would initialize the begin_to_end phase and observe subsequent state changes in the begin_to_end phase (and perhaps in the execution phase) in one RMF interval. The elapsed time of the transaction might not be recorded until a subsequent RMF interval.

These anomalies can cause response time calculations to be misleading, as discussed in Section 4. More importantly, the Workload Manager algorithms may be less effective if never-ending or long-running transactions are in the same service class as interactive transactions. This

[4]For the moment, we can ignore the time required by the SRM to assign the transaction to a CICS region, the time for the CICS region to issue the IWMCLSFY macro, the time for the Workload Manager to classify the transaction to a service class, and the time for the CICS TOR to issue the IWWMINIT macro. These times normally are very small.

is because the Workload Manager's computation of response times may be distorted by the long-running transactions.

CPExpert can identify situations when the computed number of samples is significantly different from the expected number of samples. If the begin_to_end phase sample count is significantly less than the computed number of samples, CPExpert can confidently conclude that there were long-running transactions executing in the service class, and that the long-running transactions ended in the RMF measurement interval being analyzed.

CPExpert produces Rule WLM113 when the number of begin_to_end samples is less than 50% of the number of computed samples[5]. Rule WLM113 advises you that long-running transactions executed in the service class for which you have specified a response goal.

Suggestion: CPExpert suggests that you identify the long-running transactions and remove them for the service class identified by Rule WLM110. Since the CICS transactions are long-running, it makes no sense to have the transactions in a service class with an interactive response goal.

IBM suggests the following guidance for CICS transactions:

- Do not mix CICS-supplied transactions with user transactions
- Do not mix routed with non-routed transactions
- Do not mix conversational with pseudo-conversational transactions
- Do not mix long-running and short-running transactions.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

[5]Actually, the 50% value may be overly generous. As mentioned in Footnote 4, the delay between the time MVS recognizes a transaction and the begin_to_end phase begins should be quite small. It is possible that CPExpert may use some much larger value (e.g., 90%) in the future. On the other hand, long-running transactions will likely have a **significant** effect on analysis and probably will be identified with the 50% value.

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM114: BTE Phase had large number of Ready samples

Finding: CPExpert has detected that a large percent of the begin_to_end (BTE) phase samples were in the Ready state. This finding applies only to service classes representing CICS/ESA Version 4 transactions.

Impact: This finding means that CICS transactions were waiting for dispatch in the Transaction Owning Region but were not dispatched by CICS.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

CICS provides the System Resources Manager (SRM) with information about the phase (begin_to_end or execution) of transactions by executing the IWMMINIT ("Initialize the Monitoring Environment") macro. The DURATION parameter of the IWMMINIT macro tells the SRM whether the following information related to a transaction is associated with the begin_to_end phase or with the execution phase.

The IWMMINIT macro is issued immediately after CICS has issued the IWMCLSFY ("Assigning Incoming Work Requests to a Service Class") macro to establish a service class for a transaction. Thus, the SRM quickly

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1997, Computer Management Sciences, Inc.

knows (1) the service class to which a transaction belongs and (2) whether the transaction is in its begin_to_end phase or in its execution phase.

CICS or IMS will provide the SRM with information about the state of the transaction (active state, ready state, waiting state, etc.) by issuing the IWMMCHST ("Change State of Work Request") macro. The SRM simply sets bits in a status word to indicate the state of a transaction.

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class

Included in the state reported by CICS are the times the transaction is in a Ready state. The Ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to execute in the "served" service class than were dispatched by CICS.

CICS transactions typically enter the system via a CICS TOR. The transactions receive some initial processing in the TOR and are routed to an AOR for actual application processing. CICS signals the beginning of the execution phase for the transaction when the transaction is received by the AOR.

Some transactions are not routed to an AOR, however. These transactions are completely processed in the TOR. Since the AOR signals the beginning of the execution phase, these transactions never enter the execution phase.

The service class being analyzed by CPExpert exceeded its performance objective (as reported by Rule WLM104 or Rule WLM105). Further, CPExpert had been directed to analyze response time based on the begin_to_end phase[4].

CPExpert produces Rule WLM114 when the Ready samples account for more than 25% of the number of begin_to_end samples AND when you have directed CPExpert to analyze response delays based on the begin_to_end phase. CPExpert concludes that a large percentage of non-routed transactions are processed in the service class if more than 25% of

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

[4]That is, you had specified **%LET PHASE=BEGIN_TO_END** in USOURCE(WLMGUIDE).

the transaction samples occurred in the Ready state of the begin_to_end phase.

This means that CICS tasks were waiting dispatch in the TOR, but could not be dispatched because (1) the CICS TOR was denied access to a CPU because its MVS dispatching priority was not high enough or (2) the CICS TOR was processing other CICS tasks.

The following example illustrates the output from Rule WLM114:

```
RULE WLM114:  BTE PHASE HAD LARGE READY SAMPLES

   CPExpert has detected that a large number of BEGIN_TO_END PHASE Ready
   samples were recorded for the CICS Service Class.  These Ready tasks
   would be shown as "Dispatchable" by the CEMT INQUIRE TASK command.
   This means that CICS tasks were waiting dispatch in the TOR, but could
   not be dispatched because (1) the CICS TOR was denied access to a CPU
   because its MVS dispatching priority was not high enough or (2) the CICS
   TOR was processing other CICS tasks.  Please refer to Rule WLM114 in the
   WLM Component User Manual for alternatives to correct the situation.  This
   finding applies to the following RMF measurement intervals:

                                 BEGIN TO END        READY     ACTIVE
   MEASUREMENT INTERVAL          PHASE SAMPLES      SAMPLES    SAMPLES
   10:00-10:30,26MAR1996            511,574         209,486      7,238
   10:30-11:00,26MAR1996            513,461         289,929      6,895
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- **The CICS TOR was denied access to a CPU**. Please refer to Rule WLM250 for a discussion and alternatives when a service class is denied access to a CPU.

- **The CICS TOR was processing other CICS tasks**. Please refer to Rule WLM121 for a discussion and alternatives when the CICS TOR was processing other CICS tasks.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM115: Service class did not have Begin_to_end samples

Finding: CPExpert has detected that a large percent of the begin_to_end (BTE) phase samples were in the Ready state. This finding applies only to service classes representing CICS/ESA Version 4 transactions.

Impact: This finding means that CICS transactions were waiting for dispatch in the Transaction Owning Region but were not dispatched by CICS.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

CICS transactions typically enter the system via a CICS TOR. The transactions receive some initial processing in the TOR and are routed to an AOR for actual application processing. CICS signals the beginning of the execution phase for the transaction when the transaction is received by the AOR.

The AOR to which the transaction is routed can reside on the system on which the TOR resides, or the AOR can reside on another system in the sysplex. If the AOR resides on the same system as the TOR, SMF data on the system will reflect both the begin_to_end phase information and the

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1997, Computer Management Sciences, Inc.

execution phase information. However, if the AOR resides on a different system in the sysplex, SMF data from that system will not reflect begin_to_end phase information for the transaction.

CPExpert detected that a transaction service class exceeded its response goal on at least one system in the sysplex being analyzed. However, there were no begin_to_end samples describing the service class on the local system. Consequently, CPExpert analyzes the Execution Phase on the local system. CPExpert produces delay-related information based on the Execution Phase, and produces delay-related information for the server(s) on the local system providing service to the transaction service class.

CPExpert produces Rule WLM115 to provide information about the delays on the local system. The following example illustrates the output from Rule WLM115:

```
RULE WLM115:  SERVICE CLASS DID NOT HAVE BEGIN_TO_END PHASE SAMPLES

   CICS is a "served" Service Class (e.g., IMS or CICS transactions).
   However, this service class did not have any Begin_to_End samples on
   System J80, while the service class had a number of ended transactions
   in the Execution Phase.  Further, the CICS Service Class missed its
   performance goal on at least one other system.  CPExpert assumes that
   the transactions have been shipped from another system to System J80.
   CPExpert analyzed delays to the CICS Service Class for measurement
   intervals in which the service class missed its performance goal on
   another system.  CICS was served by CICSRGN.

                             TOTAL   AVERAGE TIME IN   PRIMARY,SECONDARY
   MEASUREMENT INTERVAL      TRANS   EXECUTION PHASE   CAUSES OF DELAY
   13:00-13:30,26MAR1996    87,239         0.711       ACTIVE(44%),WAIT I/O(38%)
```

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules may be produced to provide more information. Please refer to Rule WLM104 or Rule WLM105 for information about the causes of delay to the subsystem transaction service classes.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM116: Execution Phase samples did not exist in SMF data

Finding: CPExpert has detected that there were no Work Manager/Resource Manager sample in the Execution Phase. This finding applies only to service classes representing transactions for CICS/ESA Version 4 and subsequent versions of CICS.

Impact: This finding means that all service class activity for the indicated RMF intervals took place in the Begin_to_end phase.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: Beginning with CICS/ESA Version 4.1, CICS reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[1].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 or CICS/TS has classified the transaction[2]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1, CICS/TS, or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

The SRM periodically samples the status word associated with each transaction[3], and updates counters representing the state of transactions executing in the service class. There is a status word for the begin_to_end phase and a status word for the execution phase, and separate sets of counters are maintained for the various begin_to_end states and execution states for each service class The result of the sampling is recorded in SMF Type 72 records, as the Work Manager/Resource Manager section.

---

[1]IMS Version 5 reports only *execution phase* samples.

[2]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

[3]With MVS/ESA SP5.1, the SRM takes its samples every 250 milliseconds.

©Copyright 1994, Computer Management Sciences, Inc.

There are separate record sections for the Begin_to_end phase and the Execution phase.

The service class being analyzed by CPExpert did not meet its performance goal (as reported by Rule WLM104 or Rule WLM105). However, the SMF data did not contain samples in the Execution phase section of the Work Manager/Resource Manager information.

CPExpert produces Rule WLM116 when analyzing performance of the service class from the perspective of Execution phase, and there were no Execution phase samples. Since CPExpert is analyzing response delays based on Execution phase samples, Rule WLM116 advises you that the analysis cannot be performed because there were no Execution phase samples. Rule WLM116 provides information about transactions ending in the Begin_to_end phase, and a distribution of the percent of samples that were in each major state.

The following example illustrates the output from Rule WLM105, leading to Rule WLM116:

```
RULE WLM105: SERVICE CLASS DID NOT ACHIEVE PERCENTILE RESPONSE GOAL

   CICSCONV: Service class did not achieve its response goal during the
   measurement intervals shown below.  The response goal was 90.00 percent
   of the transactions completing within 1.000 seconds, with an importance
   level of 3.  CICSCONV was defined as a "served" Service Class (e.g.,
   IMS or CICS transactions).  The below causes of delay were based upon
   local Execution Phase samples.
   CICSCONV was served by CICSRGN

                            -------LOCAL SYSTEM--------
                                    TRANS      %
                            TOTAL MEETING  MEETING PERF PLEX PRIMARY,SECOND
   MEASUREMENT INTERVAL     TRANS   GOAL     GOAL  INDX  PI  CAUSE OF DELAY
   11:45-12:00,18MAR1998       81       0      0.0  4.00 4.00  NO EXE PHASE SAMPLES
   12:00-12:15,18MAR1998       89       1      1.1  4.00 4.00  NO EXE PHASE SAMPLES



RULE WLM116: SERVICE CLASS DID NOT HAVE EXECUTION PHASE SAMPLES

   CICSCONV is a "served" Service Class (e.g., IMS or CICS transactions).
   However, this service class did not have any Execution Phase samples
   on System J80 during the intervals shown below.  The below information
   shows the total samples collected and the distribution of samples in
   the Begin_to_End Phase for CICSCONV:

                            TOTAL   TOTAL  ------PERCENT OF SAMPLES-------
   MEASUREMENT INTERVAL     TRANS  SAMPLES IDLE READY ACTIVE WAIT SWITCHED
   11:45-12:00,18MAR1998   2,317      81  99.7   0.1   0.3   0.0    0.0
   12:00-12:15,18MAR1998   3,056      89  99.9   0.0   0.1   0.0    0.0
```

Suggestion: The situation in which a transaction service class misses its performance goal, but there are no Execution phase samples normally is caused by the following situations:

- Transactions in the service class on the system being analyzed complete in the Begin_to_end phase, and they are not shipped to an AOR. These transactions commonly are CICS system transactions. The example shown above illustrates this situation. Note that a relatively small number of transactions completed execution, and that the transactions were Idle during a large percent of the samples. In such cases, you may wish to ignore CPExpert's finding, or change the guidance in USOURCE(WLMGUIDE) to exclude the service class from analysis.

  At present, CPExpert does no further analysis of the Begin_to_end phase delays. This design is because all situations encountered had (1) few transactions involved, and (2) most of the samples were in Idle state. Please call if you encounter situations that you feel should be analyzed further.

- All transactions in the service class being analyzed are shipped to another system in the sysplex.

  In this situation, CPExpert will "set a flag" and analyze Execution phase activity for the service class on other systems in the sysplex.

- All transactions in the service class being analyzed are shipped somewhere in the network.

  In this situation, no further information is available in SMF, and no further analysis can be done.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.1 Performance Guide
Section 2.6.3.1: Service Definitions

CICS/TS Release 1.2 Performance Guide
Section 2.6.3.1: Service Definitions

©Copyright 1994, Computer Management Sciences, Inc.

CICS/TS Release 1.3 Performance Guide
Section 2.5.7.1: Service Definitions

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads - Setting up service definitions).

Rule WLM119: Work Manager did not collect data for service class

Finding: The subsystem work manager did not collect delay data for the service classes "served" by the work manager. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has NO IMPACT on performance of your computer system. The finding is provided simply to explain why CPExpert cannot analyze delay information for the "served" service class which has missed its service goal.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105, the logic of these rules tries to identify the cause of the delay, from the "served" service class view.

If the subsystem supports work manager delay reporting, the information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records.

If the subsystem does not support work manager delay reporting, the information is not available, and CPExpert cannot identify the cause of the delay.

The following example illustrates the output from Rule WLM119:

```
RULE WLM119:  WORK MANAGER DID NOT COLLECT DATA FOR SERVICE CLASS

   The subsystem work manager did not collect delay data for the IMS
   Service Class.  Consequently, detailed data about transaction delays
   is not available for CPExpert to analyze.  CPExpert will analyze the
   "server" Service Class data in an attempt to identify why IMS did
   not meet its service goal.
```

©Copyright 1994, Computer Management Sciences, Inc.

Suggestion: There are no suggestions with this finding, since it simply explains why CPExpert cannot provide primary and secondary causes of delay for the service class missing its service goal. CPExpert will analyze the "server" service class and other rules will be produced to provide more information.

Rule WLM120: Significant transaction time was in Active state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Active state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Active state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Ready state**.
- **Active state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

CPExpert produces Rule WLM120 when the primary or secondary cause of delay was that the transaction service class was in the Active state for a significant percent of its response time.

The Active state indicates that a task was executing on behalf of the transaction, **from the perspective of CICS or IMS**. This last phrase is in bold to indicate that the information is only from the perspective of CICS or IMS.

The transaction is not active, of course, even though the Active state is reported for the transaction service class. The actual "Active state" is the state of the **task associated with the transaction**. For CICS transactions, this is the time accounted for by tasks executing in the CICS region. These tasks would be shown as "Running" by the CEMT INQUIRE TASK command.

The fact that CICS reports "Active" state does not mean that the CICS programs are actually processing the transaction. MVS allocates CPU cycles based on dispatching priority, and the CICS region may be denied access to a CPU.

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

CICS might have dispatched a task from the dispatch queue. However, the task could be preempted by other address spaces outside of CICS. For example, an address space with a higher dispatching priority could have preempted CICS. Consequently, CICS could be waiting for access to a CPU and not actually executing, although the CICS region would have reported to the Workload Manager that the transaction was in Active state.

The following example illustrates the output from Rule WLM120:

```
RULE WLM120:  SIGNIFICANT TRANSACTION TIME WAS IN ACTIVE STATE

  A significant amount of the transaction response time for CICSAMP Service
  Class was spent in the Active State.  For CICS transactions, this is the
  time accounted for by tasks executing in the CICS region.  These tasks
  would be shown as "Running" by the CEMT INQUIRE TASK command.  The fact
  that CICS reports "Active" state does not mean that the CICS programs
  are actually processing the transaction, however.  MVS allocates CPU
  cycles based on dispatching priority, and the CICS region may be denied
  access to a CPU.  CPExpert will analyze the CICS regions to determine
  whether the regions were denied access to a CPU.
```

Suggestion: There are no suggestions directly associated with this finding. The tasks supporting the transaction service class are active from the perspective of CICS. Actions to improve performance depend upon whether the server service class is actually using the CPU or whether the server service class is denied use of the CPU.

- **Using the CPU**. If the server service class is primarily using the CPU, actions could be taken to optimize application code of the tasks serving the transactions. These actions should reduce the CPU requirements of the code. Alternatively, performance improvement actions could include increasing the CPU capacity by acquiring a faster processor.

- **Denied use of the CPU**. If the server service class is denied use of the CPU, actions could be taken to increase the relative CPU dispatching priority of the server service class.

  In Goal Mode, users cannot specify a dispatching priority for address spaces or service classes. The Workload Manager adjusts dispatching priority based upon the importance of performance goals associated with the service class and based on whether the service class is meeting its performance goal. By definition, the service class identified by this rule is not meeting its performance goal.

Consequently, a user can affect the relative CPU dispatching priority only by (1) increasing the goal importance of the transaction service class or (2) decreasing the goal importance of other service classes.

CPExpert will analyze the "server" service class to determine whether the server (e.g., the CICS region) was using the CPU, or whether the server was denied access to the CPU. As a result of CPExpert's analysis, other rules may be produced to provide more information.

Rule WLM121: Significant transaction time was in Ready state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Ready state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has a MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Ready state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Ready state**.
- **Active state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

CPExpert produces Rule WLM121 when the primary or secondary cause of delay was that the transaction service class was in the Ready state for a significant percent of its response time. The Ready state indicates that a task associated with the transaction was ready to execute, but was not selected by the work manager.

For CICS transactions, this is the time accounted for by tasks executing in the CICS region. These tasks would be shown as "Dispatchable" by the CEMT INQUIRE TASK command.

The following example illustrates the output from Rule WLM121:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM121:  SIGNIFICANT TRANSACTION TIME WAS IN READY STATE

   A significant amount of the transaction response time for CICUSRTX Service
   Class was spent in the Ready State.  For CICS transactions, this is the
   time accounted for by tasks which were not executing in the CICS region,
   but which were ready to be dispatched.  The tasks were not dispatched
   because CICS had given priority to another task.  These tasks would be
   shown as "Dispatchable" by the CEMT INQUIRE TASK command.  If this
   finding is consistently made for an important service class, you may
   wish to consider (1) investigating the long-running tasks which ARE
   being dispatched, (2) adjusting the priority which CICS gives to tasks,
   or (3) adding another CICS region to reduce the Ready time.
```

Suggestion: CPExpert suggests that you verify the performance goals specified for the transaction service class which has missed its performance goal.

If the performance goals for the transaction service class represent your management objectives, CPExpert suggests that you consider the following alternatives:

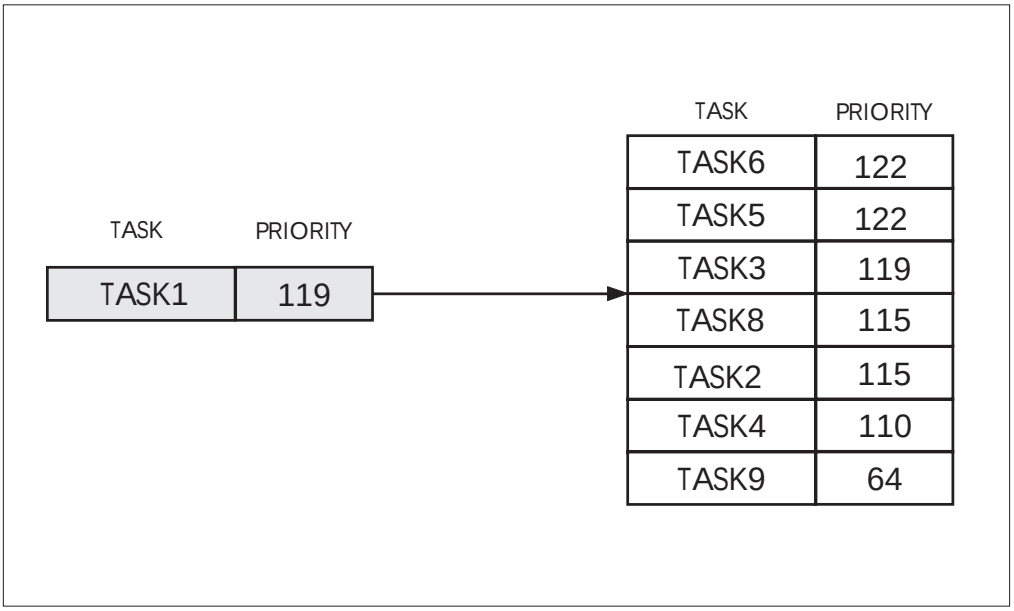
- **Review CICS task prioritization**. The task supporting the transaction service class which missed its performance goal was waiting for dispatch within the CICS region, since the CPExpert determined that the tasks spent a significant amount of time in the Ready state. One way to improve the response of important transactions is to give specific tasks preference in being dispatched by CICS.

  Dispatching priority of tasks **within a CICS region**[5] is specified in three ways: (1) priority by terminal in the CEDA TERMINAL definition (the value of the TERMPRIORITY keyword), (2) priority by transaction in the CEDA TRANSACTION definition (the value of the PRIORITY keyword), and (3) priority by operator in the signon table (the value of the OPRTY keyword in the SNT). Additionally, the three priorities can be specified via the CEMT command. The overall priority is determined by summing the priorities in the three definitions for each task, with a maximum resulting priority of 255.

  CICS maintains a dispatch queue of tasks which are ready to execute. The dispatch queue is ordered by priority, and CICS selects tasks from the top of the queue to dispatch. If task prioritization is not implemented, tasks are placed on the bottom of the queue as they become ready to execute. Thus, CICS selects tasks for dispatching in the order in which the tasks becomes ready to execute.

[5]The dispatching priority within a CICS region has no relationship to the dispatching priority from the perspective of MVS. The dispatching priority within the CICS region controls the order in which tasks are placed onto the dispatching queue in the region.

If task prioritization is implemented, a task which becomes ready to execute is placed on the queue based on its priority. A high priority task becoming ready to execute is placed on the queue ahead of all lower priority tasks, but below tasks at the same priority. The following figure illustrates the placement of tasks on the queue:



TASK1 is shown as a newly-ready task with a priority of 119. TASK1 will be placed in the CICS dispatching queue ahead of TASK8 but below TASK3.

Additionally, the dispatching priority can be increased based on the length of time a task has remained on the dispatching queue without being dispatched. The PRTYAGE parameter in the System Initialization Table (SIT) controls the frequency with which a task is examined to determine whether its priority should be increased.

The PRTYAGE specification is in milliseconds, and directs CICS to increase the priority of a task once the task has been on the dispatch queue for the PRTYAGE duration. The default value of the PRTYAGE parameter is 32768, indicating that a task's priority will be increased by 1 when the task has been on the dispatch queue for 32,768 milliseconds.

Task prioritization should be used sparingly, with task priority given to only the most important CICS tasks. The *CICS/ESA Version 4.1 Performance Guide* (Section 4.7.6 - Task Prioritization) explains the effects, uses, limitations, and implementation of task prioritization.

- **Remove selected transactions from the CICS region**. CICS task prioritization is not interrupt driven, as is MVS dispatching. The CICS task prioritization scheme simply relates to the relative position of tasks on the CICS dispatching queue. Once CICS has selected a task for dispatching, the task will remain dispatched until the task returns control to CICS.

  The Workload Manager allocates resources to address spaces (e.g., CICS regions), not to transaction service classes. The CICS region could be providing good service to other, less important transactions in different service classes. These service classes could be using significant system resources and delaying CICS in its dispatching the important transactions.

  If you have relatively long-running tasks serving transactions with a relatively low importance, these tasks may retain control of CICS for prolonged intervals. The result may be that the transactions with a high importance are delayed waiting for CICS to select their corresponding tasks for dispatch. One further result may be that the Workload Manager may allocate more resources to the service class representing the CICS region. Unfortunately, the additional resources may not help improve performance of the important tasks since CICS internally controls dispatching of tasks and these tasks may not release control.

  The only solution to this problem may be to identify the long-running transactions and remove them from the CICS region altogether. In general, this would be the preferred approach (that is, it normally is preferable to have a CICS region serving only your most important transactions.) This approach may require that another CICS region be generated, however.

- **Identify "long" transactions and optimize their related tasks**. This approach may result in large benefits, but generally requires a significant amount of application programmer time.

- **Speed the flow of all transactions through the CICS region**. The CICS region operates within the standard MVS environment. The CICS region may be delayed for various reasons (CPU dispatching, I/O access, etc.). CPExpert will analyze the "server" service class to determine whether the server (i.e., the CICS region) was using the CPU, whether the server was denied access to the CPU, etc. As a result of CPExpert's analysis, other rules may be produced to provide more information.

Rule WLM122: Significant transaction time was in Idle state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Idle state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has NO IMPACT, LOW IMPACT, MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The finding primarily indicates that either (1) the workload classification scheme improperly groups conversational transactions in the same service class as non-conversational transactions or (2) the performance goal has been improperly specified for the service class.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

CPExpert produces Rule WLM122 when the primary or secondary cause of delay was that the transaction service class was in the Idle state for a significant percent of its response time. The Idle state indicates that no work request was available to the work manager (CICS or IMS) that is allowed to run.

For CICS transactions, this is the time accounted for by tasks executing in the CICS region. These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK command.

For CICS transactions, this time differs depending upon the types of tasks executing.

- Tasks could be waiting of a principal facility (for example, conversational tasks which were waiting for a resource from a terminal user).
- The Terminal Control (TC) task (CSTP) could be waiting for work.

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

- The interregion controller task (CSNC) could be waiting for transaction routing requests.

- CICS system tasks (such as CSSY) could be waiting for work.

None of these tasks should be in a service class with a response goal, as neither CICS nor the Workload Manager can provide resources to reduce the response time.

The following example illustrates the output from Rule WLM122:

```
RULE WLM122:  SIGNIFICANT TRANSACTION TIME WAS IN IDLE STATE

  A significant amount of the transaction response time for CICUSRA Service
  Class was spent in the Idle State.  For CICS transactions, this time
  differs depending upon the types of tasks executing.
    - Tasks could be waiting of a principal facility (for example,
      conversational tasks which were waiting for a resource from a
      terminal user).
    - The Terminal Control (TC) task (CSTP) could be waiting for work.
    - The interregion controller task (CSNC) could be waiting for
      transaction routing requests.
    - CICS system tasks (such as CSSY) could be waiting for work.
  These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK
  command.  CPExpert suggests that these transactions be identified and
  placed into their own service class.  Idle time normally should not
  included in a service class with response performance objectives.
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- **Modify your workload classification scheme**. The most likely problem is that the workload classification scheme does not adequately partition the transactions into time-critical service classes and service classes which do not have a critical response goal.

  CPExpert suggests that you modify the workload classification scheme such that the transactions experiencing Idle state time are placed into a service class different from the service class containing important transactions. While it may be true that the transactions experiencing Idle state time are "important" transactions, the Workload Manager cannot allocate resources to reduce response for transactions which are Idle state for reasons outside the Workload Manager's control.

- **Review the performance goal for the service class**. From a "conceptual" view, the transactions experiencing Idle state "should" be assigned an execution velocity goal; they would receive CPU time when they wanted the CPU time. Unfortunately, the Workload Manager cannot

assign resources to transactions, but assigns the resources to address spaces supporting the transactions. Thus, the Workload Manager ISPF application does not allow transaction subsystem service classes to be defined with any goal other than a response goal.

If you specify a short response goal, the Workload Manager will incur overhead attempting to meet a performance goal for events outside its control. While the Workload Manager often will detect this situation (that is, it will detect that it cannot take action to improve response for the service class), there is no point in having the Workload Manager incur the overhead required to make the decision.

CPExpert suggests that you specify a **very long** response goal[5] for the service class containing the transactions in Idle state. These transactions are idle (Suspended) waiting for events outside the Workload Manager's control.

This action should be done only after important transactions with valid response goals have been removed from the service class! You should modify your workload classification scheme, if necessary, to make sure that the important transactions have been removed from the service class with the long response goal.

[5]CPExpert identifies transaction subsystem service classes and will suppress Rule WLM006 for these service classes.

Rule WLM123: Significant transaction time was in Waiting for Lock state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Lock state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Lock state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

---

Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

IMS Version 5 reports only *execution phase* samples.

Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CPExpert produces Rule WLM123 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Lock state for a significant percent of its response time.

For CICS transactions, this is the time accounted for by tasks which were suspended waiting for such locks as:

- A lock on a CICS resource.
- A record lock on a recoverable VSAM file.
- Exclusive control of a record in a BDAM file

Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

- An application resource that has been locked by an EXEC CICS ENQ command

These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK command.

The following example illustrates the output from Rule WLM123:

```
RULE WLM123:  SIGNIFICANT TRANSACTION TIME WAS WAITING FOR LOCK

  A significant amount of the transaction response time for CICUSRB Service
  Class was spent in the Waiting for Lock State.  For CICS transactions,
  this is the time accounted for by tasks which were suspended waiting
  for such locks as:
    - A lock on a CICS resource.
    - A record lock on a recoverable VSAM file.
    - Exclusive control of a record in a BDAM file
    - An application resource that has been locked by an EXEC CICS
      ENQ command
  These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK
  command.
```

Suggestion: IBM has provided detailed information about the Workload Manager I/O Wait types used by CICS. Exhibit WLM123-1 shows the resources that a suspended task might be waiting on for the Workload Manager Lock Wait type.

Many of the causes of time spent Waiting for Lock are related to application design, and the solutions often require a review of the approach to the application or file design.

As shown in Exhibit WLM123-1, there are seven reasons that CICS provides the Workload Manager with a Wait for Lock.

TYPE OF WAIT	TYPE OF TASK	RESOURCE TYPE	RESOURCE NAME	SUSPENDING MODULE
CICS system task waits	System task	AP_TERM	STP_DONE	DFHAPDM
File control waits	User task	KC_ENQ	SUSPEND	DFHXCPC
Loader waits	User task	PROGRAM	program_ID	DFHLDLD
Lock manager waits	User task	(none)	LMQUEUE	DFHLMLM
Task control waits	User task	KCCOMPAT	CICS	DFHXCPA
Task control waits	User task	KC_ENQ	SUSPEND	DFHXCPC
Temporary storage wait	User task	TSBUFFER	(none)	DFHTSP
Temporary storage wait	User task	TSEXTEND	(none)	DFHTSP
Temporary storage wait	User task	TSOPEN4B	(none)	DFHTSP
Temporary storage wait	User task	TSQUEUE	(none)	DFHTSP
Temporary storage wait	User task	TSSTRING	(none)	DFHTSP
Temporary storage wait	User task	TSUT	(none)	DFHTSP
Temporary storage wait	User task	TSWBUFFR	(none)	DFHTSP
Transient data waits	User task	KC_ENQ	SUSPEND	DFHXCPC
Transient data waits	User task	TDEPLOCK	transient	DFHTDEXP
Transient data waits	User task	TDIPLOCK	transient	DFHTDSUB

CICS WAITING FOR LOCK
Exhibit WLM123-1

- **CICS system task waits**. CICS module DFHAPDM is the Application Domain (AP) module responsible for initializing, quiescing, and terminating the application domain. CICS provides the Workload Manger with a Wait for Lock when the application domain is being terminated (shutdown or takeover). This lock type would not cause an individual transaction to miss its performance goal.

- **File Control waits**. Lock waits caused by file control can occur when a task is waiting for a record lock in a recoverable VSAM file. When an application updates a record in a recoverable VSAM file, locking occurs at two levels: (1) VSAM locks the Control Interval (CI) when the record has been read, and (2) CICS locks the record.

  The CI lock is released as soon as the REWRITE (or UNLOCK) request is completed. However, the record is not unlocked by CICS until the updating task has reached a syncpoint. This is to ensure that data integrity is maintained if the task fails before the syncpoint and the record has to be backed out.

  If a second task attempts to update the same record while the record is still locked, the second task is suspended on resource type KC_ENQ until the lock is released. This can be a long wait, because the update might depend on a terminal operator typing in data. Also, the suspended task relinquishes its VSAM string and may relinquish its exclusive control

of the CI. The suspended task would have to regain these resouces and may have to wait after it was no longer in a file control lock wait.

BDAM does not use the "control interval" concept. When a task reads a record for update, the record is locked so that concurrent changes cannot be made by two transactions. The lock is released at the end of the current logical unit of work. If a second task attempts to update the same record while the first has the lock, it is suspended on resource type KC_ENQ.

Solving Lock Wait due to file control may require a review of the application logic or file design to see if the record-locking time can be reduced.

- **Loader waits**. A task is suspended by the loader domain if it has requested a program load and another task is already loading that program. Once the load in progress is complete, the suspended task normally is resumed quickly and the wait is unlikely to be detected.

  If the requested program is not loaded quickly, there are two likely causes:

  - **The system could be short on storage (SOS)**, so only system tasks can be dispatched. The Storage Manager Statistics part of the CICS interval statistics contain information which can be analyzed to determine whether the WLM Lock wait was likely caused by a SOS condition. The field SMSSOS is a count of the number of times CICS went SOS in a particular subpool (note that there are separate statistics for each of the storage subpools).

    - If the SMSSOS value is zero, you can be sure that the WLM Lock waits were **not** caused by Loader waits.

    - If the SMSSOS value is non-zero, it is possible that the WLM Lock waits **were** caused by Loader waits because CICS entered SOS. Unfortunately, there is no way to determine whether a task suspended for a Loader wait actually was in the service class missing its performance goal. However, the CICS region was encountering SOS, and you should take action.

      If the SMSSOS value is non-zero, CPExpert suggests that you review the suggested actions beginning on page 251 of the IBM *CICS Verson 4.1 Performance Guide*. These actions provide a checklist for reducing the virtual storage requirements above and below the 16MD line.

Alternatively, you can execute the CICS Component of CPExpert against the CICS region(s) serving the service class missing its performance goal. The CICS Component will analyze the CICS interval statistics to identify performance problems.

- **There could be an I/O error on a library**. You can check for messages that might indicate an I/O error on a library. If you find that an I/O error occurred, you should investigate the reason why the I/O error occurred.

- **Lock Manager waits**. The Lock Manager suspends a task when the task cannot acquire the lock on a resource it has requested, probably because another task has not released it. A user task cannot explicitly acquire a lock on a resource, but many of the CICS modules that run on behalf of user tasks do lock resources. Lock Manager waits could indicate a CICS system error.

  You should review the "Lock Manager Waits" part of Section 2.3: Dealing with waits (Bookmanager document) of the CICS/ESA Version 4.1 Problem Determination Guide.

  While it is possible to experience Lock Manager waits, it is unlikely that these are the cause of performance problems with the service class missing its performance goal.

- **Task Control waits**. Task Control will suspend a task (1) if the task has attempted to change the state of a file but another task is still using the file, (2) if the task attepted to update a record in a recoverable file while another task has a lock on the file, or (3) if a task has finished using a file but not issued an EXEC CICS DEQ command or a DFHKC TYPE=DEQ macro call.

  Solving these problems require a review of the approach to the application or file design.

- **Temporary storage waits**.

  - Resource type TSBUFFER indicates that the task that is waiting has issued an auxiliary temporary storage request, but the buffers are all in use. If you find that tasks are often made to wait on this resource, consider increasing the number of auxiliary temporary storage buffers (system initialization parameter TS).

  - Resource type TSEXTEND indicates that the waiting task has issued a request to extend the auxiliary temporary storage data set, but some other task has already made the same request. The wait does

not extend beyond the time taken for the extend operation to complete. If you have a task that is waiting for a long time on this resource, it is likely that there is a hardware fault or a problem with VSAM.

- Resource type TSQUEUE indicates that the waiting task has issued a request against a temporary storage queue that is already in use by another task. The latter task is said to have the lock on the queue.

  The length of time that a task has the lock on a temporary storage queue depends on whether or not the queue is recoverable. If the queue is recoverable, the task has the lock until the logical unit of work is complete. If it is not recoverable, the task has the lock for the duration of the temporary storage request only.

- Resource type TSSTRING indicates that the task is waiting for an auxiliary temporary storage VSAM string. If you find that tasks frequently wait on this resource, consider increasing the number of temporary storage strings (system initialization parameter TS).

- If a user task is waiting on resource type TSUT, activity keypointing is taking place. This involves a large amount of I/O, and, if there are many temporary storage queues, it could take a relatively long time to complete.

- Resource type TSWBUFFR indicates that the waiting task has issued an auxiliary temporary storage request, but the write buffers are all in use. You have no control over how temporary storage allocates read buffers and write buffers from the buffer pool, but if you find that tasks are often made to wait on this resource, increasing the number of auxiliary temporary storage buffers (system initialization parameter TS) should help solve the problem.

- **Transient data waits**. Transient data waits occur when a task is suspended on resource type TDEPLOCK, with a resource name corresponding to a transient data queue name. The task has issued a request against an extrapartition transient data queue, but another task is already accessing the same queue. The waiting task cannot resume until that activity is complete.

  Significant time spent in transient data waits occur becauset is necessary for a task to change TCB mode to open and close a data set. The task must relinquish control while this happens. Depending on the system loading, rel;inquishing control might take several seconds. This contributes to the wait that the second task experiences, while the second task is suspended on resource type TDEPLOCK,.

CICS uses the access method QSAM to write data to extrapartition transient data destinations. QSAM executes synchronously with tasks requesting its services. This means that any task invoking a QSAM service must wait until the QSAM processing is complete. If, for any reason, QSAM enters an extended wait, the requesting task also experiences an extended wait.

The possibility of an extended wait arises whenever QSAM attempts to access an extrapartition data set. QSAM uses the MVS RESERVE volume-locking mechanism to gain exclusive control of volumes while it accesses them, which means that any other region attempting to write to the same volume is forced to wait.

If tasks frequently get suspended on resource type TDEPLOCK, you should determine which other transactions write data to the same extrapartition destination. You might then consider redefining the extrapartition destinations in the DCT (destination control table).

You can find further guidance information about the constraints that apply to tasks writing to intrapartition destinations in the CICS Application Programming Guide. For more details of the properties of recoverable transient data queues, see the CICS Resource Definition Guide.

- Another common cause of locks on a CICS resource is the CICS shared database facility. An IMS batch job can access a local DL/I database controlled in a CICS region. Any DL/I request from the IMS batch application program is handled through the facilities of CICS instead of IMS DB.

  A shared database region contains an IMS batch application program that processes local DL/I databases, and the application program in the shared database region is scheduled by MVS job management. The job stream for the job specifies the CICS batch region controller. The shared database program uses DL/I CALLs for database references. An application program executing in a shared database region can access only the local DL/I databases that are attached to the CICS online region.

  The CICS shared database facility can greatly increase contention for a database, particularly if update operations from batch programs are involved.

  - A normal CICS task accesses and enqueues on a small number of records from a database.

- An IMS batch program may access and enqueue on all the records in the database, effectively locking up the database until the program completes[5]. If the batch jobs are update jobs, they are likely to lock out the database from online use until they finish running, which typically takes several minutes.

The following guidance is provided by IBM in the referenced *CICS Performance Guides:*

- CICS using DBCTL performs better than function shipping. Performance can be improved by replacing any database owning region (DOR) with a DBCTL owning region.

- Users accessing DL/I databases from CICS via the IMS DBCTL facility should use IMS BMPs rather than CICS shared database.

- In general, use CICS shared database only when absolutely necessary. Either try to minimize or eliminate update operations and run batch jobs during offpeak times when the system is not busy, or use IMS data sharing.

  If it is necessary to run batch update during online operations, do one of the following:

  - Run the batch update during periods of low online activity.

  - Close down the online transactions that reference the database

  - Inform users of the database that they are most likely to experience an increase in response time during the period of updating from the batch region

  - Incorporate frequent checkpoints in batch applications.

You should also review all DL/I PSBs to minimize the contention between batch and online CICS transactions and possibly increase the priority for online transactions versus the partition control task.

If batch update operations are required, use of the IMS/ESA or DL/I Checkpoint Call can free up records when they are updated, but may complicate program restart in the case of a batch program abend.

This can also greatly increase the requirements for storage in the IMS/ESA enqueue pool

Storage for the dynamic buffer may need to be increased because a large amount of backout information may have to be kept until batch program completion.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM124: Significant transaction time was in Waiting for I/O state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for I/O state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for I/O state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CPExpert produces Rule WLM124 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for I/O state for a significant percent of its response time. This is not necessarily time actually performing I/O, but could be any activity related to the I/O request. For CICS transactions, this time includes:

- File Control requests.
- Transient data requests.
- Temporary storage requests.
- Journaling I/O requests.

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

- Waiting for I/O buffers or VSAM strings.

These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK command.

The following example illustrates the output from Rule WLM124:

```
RULE WLM124:  SIGNIFICANT TRANSACTION TIME WAS WAITING FOR I/O REQUEST

   A significant amount of the transaction response time for CICUSRTX Service
   Class was spent waiting for some I/O request.  This is not necessarily
   time actually performing I/O, but could be any activity related to the
   I/O request.  For CICS transactions, this time includes:
     - File Control requests.
     - Transient data requests.
     - Temporary storage requests.
     - Journaling I/O requests.
     - Waiting for I/O buffers or VSAM strings.
   These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK
   command.  You should execute the CPExpert CICS Component against the
   regions serving the Service Class transactions to identify the cause
   of the large Wait on I/O time.
```

Suggestion: CPExpert suggests that you execute the CICS Component of CPExpert against the CICS regions serving the service class missing its performance goal. The CICS Component of CPExpert should identify problems in I/O-related areas.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM125: Significant transaction time was in Waiting for Conversation state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Conversation state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Conversation state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CPExpert produces Rule WLM125 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Conversation state for a significant percent of its response time. These tasks would be shown as "Suspended" by the CEMT INQUIRE TASK command.

The Waiting on Conversation state means that a transaction has been switched across an intersystem communication link (MRO or ISC) to another work manager.

A CICS transaction typically enters the system through a TOR and may be routed to an AOR. The Waiting on Conversation state in the TOR would include the time the transaction was switched to the AOR, plus any queue

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

time waiting for the AOR to accept the transaction and notify the Workload Manager, plus the time in the AOR processing the transaction. The Waiting on Conversation state in the TOR would terminate when the TOR received the transaction back from the AOR. All of this Waiting on Conversation time would show up in the BTE Phase of the transaction.

Most of the Waiting on Conversation state (particularly for the BTE Phase) is explained in the Switched state:

- **Switched - Local**. The transaction has been switched, across an MRO link, to another CICS region in same MVS image.

- **Switched - Sysplex**. The transaction has been switched, across an XCF/MRO link, to another CICS region in another MVS image in the sysplex.

- **Switched - Network**. The transaction has been switched, across an ISC link, to another CICS region (which may, or may not, be in the same MVS image).

The following example illustrates the output from Rule WLM125:

```
RULE WLM125:  SIGNIFICANT TRANSACTION TIME WAS WAITING FOR CONVERSATION

 A significant amount of the transaction response time for CICUSERC Service
 Class was spent waiting on a conversation between subsystems:  waiting
 on another CICS region, an IMS region, DBCTL, etc.
```

Suggestion: There are no suggestions with this rule. The finding is provided for information purposes.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM126: Significant transaction time was in Waiting for Distributed Request state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Distributed Request state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Distributed Request state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CPExpert produces Rule WLM126 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Distributed Request state for a significant percent of its response time. The following example illustrates the output from Rule WLM126:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM126:  SIGNIFICANT TRANSACTION TIME WAS WAITING, DISTRIBUTED

  A significant amount of the transaction response time for CICUSERD Service
  Class was spent waiting for some distributed request.  CICS does not
  use the distributed request function.  If this finding occurs, please
  call Computer Management Sciences, Inc. so we can investigate the cause.
```

Suggestion: CICS does not use the Distributed Request function. If this finding occurs, please call Computer Management Sciences so we can investigate the cause.

Rule WLM127: Significant transaction time was in Waiting for Local Session state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for a Local Session state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for a Local Session state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting for a session to be established with another CICS region in the same MVS image. This finding should occur only when regions are started.

CPExpert produces Rule WLM127 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for a Local Session state for a significant percent of its response time.

The following example illustrates the output from Rule WLM127:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM127:  SIGNIFICANT TRANSACTION TIME WAS WAITING, LOCAL SESSION

  A significant amount of the transaction response time for CICUSERE Service
  Class was spent waiting for the establishment of a session with another
  CICS region in the same MVS image in the sysplex.  This finding should
  occur only when regions are started.  There may be operational problems
  or CICS region integrity problems if this finding occurs at other times.
  If this finding regularly occurs, and you determine that operational
  problems are not the cause, please call Computer Management Sciences,
  Inc. so we can investigate the cause.
```

Suggestion: This finding should not occur except during intervals when CICS regions are started. Sessions normally are established for prolonged periods.

If this finding occurs for a production environment, perhaps there are operational problems or there may be CICS region integrity problems.

If you have licensed the CICS Component of CPExpert, you should run the CICS Component to analyze problems and potential problems with the CICS regions involved.

If this finding does occur for a production environment and you determine that there are no operational problems, please call Computer Management Sciences so we can investigate the cause.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM128: Significant transaction time was in Waiting for Sysplex Session state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for a Sysplex Session state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for a Sysplex Session state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting for a session to be established with another CICS region somewhere in the sysplex. This finding should occur only when regions are started.

CPExpert produces Rule WLM128 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for a Sysplex Session state for a significant percent of its response time.

The following example illustrates the output from Rule WLM128:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM128:  SIGNIFICANT TRANSACTION TIME WAS WAITING, SYSPLEX SESSION

  A significant amount of the transaction response time for CICUSERD Service
  Class was spent waiting for the establishment of a session with another
  CICS region in a different MVS image in the sysplex.  This finding should
  occur only when regions are started.  There may be operational problems
  or CICS region integrity problems if this finding occurs at other times.
  If this finding regularly occurs, and you determine that operational
  problems are not the cause, please call Computer Management Sciences,
  Inc. so we can investigate the cause.
```

Suggestion: This finding should not occur except during intervals when CICS regions are started. Sessions normally are established for prolonged periods.

If this finding occurs for a production environment, perhaps there are operational problems or there may be CICS region integrity problems.

If you have licensed the CICS Component of CPExpert, you should run the CICS Component to analyze problems and potential problems with the CICS regions involved.

If this finding does occur for a production environment and you determine that there are no operational problems, please call Computer Management Sciences so we can investigate the cause.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM129: Significant transaction time was in Waiting for Session (Network) state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Session (Network) state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Session (Network) state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting for a session to be established with another CICS region somewhere in the network. This finding should occur only when regions are started.

CPExpert produces Rule WLM129 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Session (Network) state for a significant percent of its response time.

The following example illustrates the output from Rule WLM129:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM129:  SIGNIFICANT TRANSACTION TIME WAS WAITING, NETWORK SESSION

  A significant amount of the transaction response time for CICUSERF Service
  Class was spent waiting for the establishment of an ISC (InterSystem
  Communication) session with another CICS region.  The other CICS region
  may or may not be in the same MVS image in the sysplex.  This finding
  should occur only when regions are started.  There may be operational
  problems or CICS region integrity problems if this finding occurs at
  other times.  If this finding regularly occurs, and you determine that
  operational problems are not the cause, please call Computer Management
  Sciences, Inc. so we can investigate the cause.
```

Suggestion: This finding should not occur except during intervals when CICS regions are started. Sessions normally are established for prolonged periods.

If this finding occurs for a production environment, perhaps there are operational problems or there may be CICS region integrity problems.

If you have licensed the CICS Component of CPExpert, you should run the CICS Component to analyze problems and potential problems with the CICS regions involved.

If this finding does occur for a production environment and you determine that there are no operational problems, please call Computer Management Sciences so we can investigate the cause.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM130: Significant transaction time was in Waiting for Timer state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Timer state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Timer state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting for a timer to expire or for an interval control event to complete. These timer delays normally occur when an application had issued an EXEC CICS DELAY command or EXEC CICS WAIT EVENT command.

CPExpert produces Rule WLM130 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Timer state for a significant percent of its response time.

The following example illustrates the output from Rule WLM130:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM130:  SIGNIFICANT TRANSACTION TIME WAS WAITING FOR TIMER

  A significant amount of the transaction response time for CICUSERA Service
  Class was spent waiting for a timer event or an interval control event
  to complete.  For example, an application had issued an EXEC CICS DELAY
  or EXEC CICS WAIT EVENT command.  If this finding occurs often, CPExpert
  suggests that these transactions be identified and placed into their
  own service class.  Tasks which spend a significant amount of time
  waiting for timer expiration normally should not be included in a
  service class with response performance objectives.
```

Suggestion: If this finding occurs often, CPExpert suggests that you consider the following alternatives:

- Identify the transactions which cause the Wait for Timer delay. You should consider placing these transactions into their own service class, as it usually is inappropriate for transactions which wait for a timer to be in a service class with other transactions.

- Alternatively, you may wish to review the performance goal associated with these transactions. It is possible that the transactions have been placed into their own service class, but the performance goal associated with the service class does not adequately account for the timer delays. Since timer delays are typically an application-related function, you may wish to revise the performance goal to account for longer delays.

- Alternatively, the applications may have issued a timer delay because of the unavailability of some CICS resource. You may wish to review the application to determine the cause of the timer delay and whether the delay can be reduced.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM131: Significant transaction time was in Waiting for Another Product state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting for Another Product state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting for Another Product state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting for another product. The information provided by RMF does not identify the other product, but the product usually is DBCTL or DB2.

CPExpert produces Rule WLM131 when the primary or secondary cause of delay was that the transaction service class was in the Waiting for Another Product state for a significant percent of its response time.

The following example illustrates the output from Rule WLM131:

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

```
RULE WLM131:  SIGNIFICANT TRANSACTION TIME WAS WAITING, ANOTHER PRODUCT

  A significant amount of the transaction response time for CICSAMP Service
  Class was spent waiting for another product.  The information provided
  by RMF does not identify the other product, but the product usually is
  DBCTL or DB2.  If this finding regularly occurs, you may wish to review
  the products used by these CICS tasks to determine whether their delays
  can be reduced.
```

Suggestion: If this finding occurs often, CPExpert suggests that you review the products used by the service class. These products typically will be DBCTL or DB2. If the delay is significant, you may be able to achieve the performance goals for the service class only if the performance of the other product can be improved.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section
Section 2.7.1.2: The state section

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section
Section 2.6.1.2: The state section

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 8 (Managing Workloads).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 8 (Managing Workloads).

Rule WLM132: Significant transaction time was in Waiting (Miscellaneous) state

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent in the Waiting (Miscellaneous) state. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent in the Waiting (Miscellaneous) state.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1994, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Wait state indicates that a task in support of the transaction was waiting on some activity. The Wait state is broken into several categories: waiting for lock, waiting for I/O, waiting for conversation, waiting for distributed request, waiting for a session to be established (locally, somewhere in the network, or somewhere in the sysplex), waiting for a timer, waiting for another product, or waiting for an unidentified resource.

CICS reports the time when a work unit (that is, a task in support of a transaction) was waiting, broken into ten separate categories. Nine of the waiting categories are specific (e.g., Waiting for I/O). The tenth category is the "Miscellaneous Wait" category, used when CICS does not identify the specific reason for the wait delay.

The initial versions of CICS documentation simply described the "Miscellaneous Wait" category as being wait for unidentified reasons. In revisions to the documents, IBM has provided detailed information about the Workload Manager Miscellaneous Wait types used by CICS.

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

CPExpert produces Rule WLM132 when the primary or secondary cause of delay was that the transaction service class was in the Waiting (Miscellaneous) state for a significant percent of its response time.

The following example illustrates the output from Rule WLM132:

```
RULE WLM132:  SIGNIFICANT TRANSACTION TIME WAS WAITING, MISCELLANEOUS

   A significant amount of the transaction response time for the CICSPROD
   Service Class was spent waiting for reasons which were not identified
   by CICS.  Please refer to the description of Rule WLM132 for a discussion
   of the CICS Miscellaneous Wait categories, how to determine which CICS
   Miscellaneous Waits occur on your system, and how to reduce these waits.
```

Suggestion: IBM has provided detailed information about the Workload Manager Miscellaneous Wait types used by CICS. Exhibit WLM132-1 shows the resources that a suspended task might be waiting on for the Workload Manager Miscellaneous Wait type.

As shown in Exhibit WLM132-1, there are twelve reasons that CICS provides the Workload Manager with a Miscellaneous Wait.

- **CICS system task waits**. CICS system task waits occur (1) as a natural result of the CICS system tasks or (2) because of a system error preventing the system task from resuming.

  - Many system tasks enter a wait state as a natural result of their operation.

    - For example, the DFHSMSY module of the storage manager domain might stay suspended for a prolonged time (i.e., minutes, or even hours). The purpose of the DFHSMSY module is to clean up storage when significant changes occur in the amount being used. This situation would happen infrequently in a production system running well within its planned capacity, but the situation can occur.

    - Some system tasks perform many I/O operations. These I/O operations are subject to I/O constraints such as string availability, and volume and data set locking. In the case of tape volumes, the tasks can also be dependent on operator action while new volumes are mounted.

TYPE OF WAIT	TYPE OF TASK	RESOURCE TYPE	RESOURCE NAME	SUSPENDING MODULE
CICS system task waits	System task	(none)	DMWTQUEU	DFHDMWQ
CICS system task waits	System task	AP_INIT	CSADLECB	DFHSII1
CICS system task waits	System task	AP_INIT	ECBTCP	DFHAPSIP
CICS system task waits	System task	AP_INIT	SIPDMTEC	DFHAPSIP
CICS system task waits	System task	AP_INIT	TCTVCECB	DFHSII1
CICS system task waits	System task	AP_QUIES	CSASSI2	DFHSTP
CICS system task waits	System task	AP_QUIES	SHUTECB	DFHSTP
CICS system task waits	System task	DBDXEOT	(none)	DFHDXSTM
CICS system task waits	System task	DBDXINT	(none)	DFHXSTM
CICS system task waits	System task	DFHAIIN	AITM	DFHAIIN1
CICS system task waits	System task	DFHCPIN	CPI	DFHCPIN1
CICS system task waits	System task	DFHPRIN	PRM	DFHPRIN1
CICS system task waits	System task	DFHSIPLT	EARLYPLT	DFHSII1
CICS system task waits	System task	FCINWAIT	STATIC	DFHFCIN1
CICS system task waits	System task	JCINITN	JOURNALS	DFHJCP
CICS system task waits	System task	STARTUP	TSMCPECB	DFHRCRP
CICS system task waits	System task	SUBTASK	SISUBECB	DFHRCRP
CICS system task waits	System task	SUCNSOLE	WTO	DFHSUWT
CICS system task waits	System task	TCP_SHUT	DFHZDSP	DFHZDSP
EDF waits	User task	EDF	DBUGUSER	DFHEDFX
Front End Programming waits	User task	ADAPTER	FEPI_RQE	DFHSZATR
Front End Programming waits	CSZI	FEPRM	SZRDP	DFHSZRDP
Interval control waits	User task	ICGTWAIT	terminal_ID	DFHICP
Interval control waits	User task	ICWAIT	terminal_ID	DFHICP
Journal control waits	User task	JASUBTAS	JASTMECB	DFHJCSDJ
Journal control waits	User task	JCBUFFER	JCTBAECB	DFHJCSDJ
Journal control waits	User task	JCDETACH	SUBTASK	DFHJCSDJ
Journal control waits	User task	JCREADY	JCTXAECB	DFHJCO
Journal control waits	User task	JCREADY	JCTXBECB	DFHJCO
Journal control waits	User task	JCREADY	JCTXXECB	DFHJCO
Storage waits	User task	CDSA	(none)	DFHSMSQ
Storage waits	User task	ECDSA	(none)	DFHSMSQ
Storage waits	User task	ERDSA	(none)	DFHSMSQ
Storage waits	User task	ESDSA	(none)	DFHSMSQ
Storage waits	User task	ESDSA	(none)	DFHSMSQ
Storage waits	User task	EUDSA	(none)	DFHSMSQ
Storage waits	User task	RDSA	(none)	DFHSMSQ
Storage waits	User task	SDSA	(none)	DFHSMSQ
Storage waits	User task	UDSA	(none)	DFHSMSQ
Task control waits	User task	EKCWAIT	SINGLE	DFHEKC
Task control waits	User task	KCCOMPAT	LIST	DFHXCPA
Task control waits	User task	KCCOMPAT	SINGLE	DFHXCPA
Task control waits	User task	KCCOMPAT	SUSPEND	DFHXCPA
Task control waits	User task	KCCOMPAT	TERMINAL	DFHXCPA
Temporary storage wait	User task	TSAUX	(none)	DFHTSP
Transient data waits	User task	TD_INIT	DCT	DFHTDA
User waits	User task	FOREVER	DFHXMTA	DFHXMTA
User waits	User task	USERWAIT	ECB	list
VTAM waits	User task	ZCIOWAIT	DFHZARER	DFHZARER
VTAM waits	User task	ZCZGET	DFHZARL2	DFHZARL
VTAM waits	User task	ZCZNAC	DFHZARL3	DFHZARL
XRF waits	User task	XRPUTMSG	message_Q	DFHWMQP

CICS MISCELLANEOUS WAITS
Exhibit WLM132-1

You should consider placing CICS system tasks into a single service class. IBM suggests that you not mix CICS-supplied transactions in a service class with user transactions.

- You should contact your IBM support center if a system task is in a wait state, and there is a system error preventing it from resuming.

- **Execution Diagnostic Facility (EDF) waits**. The EDF waits are a natural result of using the Execution Diagnostic Facility.

  The EDF waits should not occur in a CICS production region. EDF waits would not be a cause for concern in a CICS test region, as they are programmer-generated.

- **Front End Programming waits**. There are two types of Front End Programming waits from the view of CICS: (1) a wait for the FEPI_RQE resource and (2) a wait for the SCRDP resource.

  - The wait for the FEPI_RQE resource is issued in the FEPI adapter when a FEPI command is passed to the Resource Manager for processing. The Wait ends when the Resource Manager has processed the request. It is possible for a FEPI_RQE wait to be outstanding for a long time (for example, when awaiting a flow from the back-end system that is delayed due to network traffic). IBM recommends that you not cancel tasks that are waiting at this point; to do so could lead to severe application problems.

  - The wait for the SCRDP resource is issued by the CSZI task in the FEPI Resource Manager when it has no work to do. The wait ends when work arrives (from either the FEPI adapter or a VTAM exit).

    An SZRDP wait is generated when the FEPI Resource Manager is idle. Consequently, the SZ TCB is also inactive. On lightly loaded systems, this occurs frequently.

  The Dispatcher Domain Statistics part of the CICS interval statistics contain information which can be analyzed to determine whether the WLM Miscellaneous Wait was likely caused by a Front End Programming wait. There are Dispatcher Domain Statistics for each TCB; TCB 4 is the secondary LU TCB and is present if FEPI=YES was specified in the System Initialization Table. Within TCB 4 statistics, the DSGTWT field holds the accumulated real time that the CICS region was in a MVS wait for the Front End Programming TCB.

  If the DSGTWT value is small, you can be reasonably sure that the WLM Miscellaneous waits were **not** caused by Front End Programming waits.

If the DSGTWT value is relatively large, it is possible that the WLM Miscellaneous waits **were** caused by Front End Programming waits. Unfortunately, there is no way to determine whether a task suspended for a Front End Programming Wait actually was in the service class missing its performance goal. However, **some** tasks in the CICS region are encountering Front End Programming Waits if the DSGTWT value is relatively large and you may wish to take action.

The CICS/ESA Front End Programming Interface User Guide (see References) should be consulted regarding improving the performance of the Front End Programming interface.

Additionally, you should consider placing CICS system tasks into a single service class. IBM suggests that you not mix CICS-supplied transactions in a service class with user transactions.

- **Interval Control waits**. Interval Control waits are caused by user tasks.

  You should review the "Interval Control Waits" part of Section 2.3: Dealing with waits (Bookmanager document) of the CICS/ESA Version 4.1 Problem Determination Guide.

- **Journal Control waits**. CICS Journal Control provides the Workload Manager with a Miscellaneous Wait for four resource types: JASUBTAS, JCBUFFER, JCDETACH, and JCREADY.

  - **JASUBTAS**. The purpose of the wait for the JASUBTAS resource is to delay shutdown until the JASP subtask has completely submitted all the archiving jobs of those journals needing to be archived.

  - **JCBUFFER**. If the resource type is JCBUFFER, with resource name JCTBAECB, the task that has requested shutdown is waiting for the journaling task to flush the buffer, close the journal, and terminate itself.

  - **JCDETACH**: A task that has requested shutdown can be made to wait on the detaching of the journal subtask from the operating system.

  - **JCREADY**. Workload Manager Miscellaneous Waits for the JCREADY resource type occur during archiving. CICS writes to a second data set while archiving the first data set either tape or disk. The first data set is not reused until archiving is complete and the operator has responded to message DFHJC4583. If the operator has not responded before the second journal data set is full, the JCT PAUSE option causes logging to cease until the operator has

responded. User tasks are made to wait on resource type JCREADY when no operator reply has been received to message DFHJC4583, and message DFHJC4584 has subsequently been issued.

Workload Manager Miscellaneous Waits for the first three Journal Control resource types occur only during shutdown, and should not cause a service class to miss its performance goal.

Workload Manager Miscellaneous Waits for the JCREADY resource type could cause serious performance problems if the operator does not respond to message DFHJC4583 in a timely manner.

The Journal Control Statistics part of the CICS interval statistics contain information which can be analyzed to determine whether the WLM Miscellaneous Wait was likely caused by CICS having to wait for the archive job. The field A13WAC is a count of the number of times CICS had to wait for a particular journal because the archive job had not completed at the time it was needed.

- If the A13WAC field is zero, you can be sure that the WLM Miscellaneous waits were **not** caused by Journal Control archiving.

- If the A13WAC value is non-zero, CPExpert suggests that you determine why message DFHJC4583 was not responded to in a timely manner. While it is uncertain that the operator response caused problems with the service class missing its performance goal, tasks are suspended because of archiving problems. You should take action to correct the problem.

Alternatively, you can execute the CICS Component of CPExpert against the CICS region(s) serving the service class missing its performance goal. The CICS Component will analyze the CICS interval statistics to identify performance problems.

- **Storage waits**. Storage waits occur when a task is waiting for any of the resource types CDSA, UDSA, ECDSA, EUDSA, ERDSA, SDSA, ESDSA, or RDSA. Waits on these resources occur when tasks make unconditional storage requests (SUSPEND=YES) that cannot be satisfied[5]. Storage requests below the 16MB line wait for CDSA, UDSA, SDSA, or RDSA. Storage requests above the line 16MB line wait for ECDSA, EUDSA, ESDSA, or ERDSA.

[5]Note that, if conditional requests are made (SUSPEND=NO), tasks are not suspended on these resources, and a miscellaneous wait would not be provided to the Workload Manager.

CICS automatically takes steps to relieve storage when it is under stress. For example, CICS would release storage occupied by programs whose current use count is zero.

The most likely reasons for extended waits on storage requests are:

- The task has issued an unconditional GETMAIN request for an unreasonably large amount of storage.

- The task has issued an unconditional GETMAIN request for a reasonable amount of storage, but the CICS region is approaching a short-on-storage (SOS) condition.

- The task has issued an unconditional GETMAIN request for a reasonable amount of storage, but storage in the CICS region could have become too fragmented for the request to be satisfied.

The Storage Manager Statistics part of the CICS interval statistics contain information which can be analyzed to determine whether the WLM Miscellaneous Wait was likely caused by a storage wait. The field SMSUCSS is a count of the number of times a task was suspended because of insufficient storage to satisfy the request at the moment.

- If the SMSUCSS value is zero, you can be sure that the WLM Miscellaneous waits were **not** caused by storage waits.

- If the SMSUCSS value is non-zero, it is possible that the WLM YMiscellaneous waits **were** caused by storage waits. Unfortunately, there is no way to determine whether a task suspended for storage constraint actually was in the service class missing its performance goal. However, tasks in the CICS region are encountering waits for storage if the SMSUCSS value is non-zero, and you should normally consider action. Further, the waiting task may be automatically purged[6] if it has waited for storage longer than the deadlock time-out parameter specified in the installed transaction definition.

  If the SMSUCSS value is non-zero, CPExpert suggests that you review the suggested actions beginning on page 171 of the IBM *CICS Verson 4.1 Performance Guide*. These actions provide a checklist for reducing the virtual storage requirements above and below the 16MD line.

Alternatively, you can execute the CICS Component of CPExpert against the CICS region(s) serving the service class missing its performance

[6]Certain conditions prevent purging of a task (as examples, a deadlock time-out value of 0, or a specification of SPURGE(NO)).

goal. The CICS Component will analyze the CICS interval statistics to identify performance problems.

- **Task Control waits**. The CICS Transaction Manager provides the Workload Manager with a Miscellaneous Wait when a task is waiting on a resource type of KCCOMPAT, and the task has been suspended by the Transaction Manager. Additionally, CICS Task Control provides the Workload Manager with a Miscellaneous Wait when a task is waiting on a resource type of EKCWAIT and has been suspended by Task Control.

  - The Miscellaneous Wait type is issued by the Transaction Manager when the task is suspended after issuing one of three macro calls:

    - A DFHKC TYPE=WAIT,DCI=LIST macro call was issued. The task is waiting for any ECB in a list of ECBs to be posted, after which the task may be resumed.

    - A DFHKC TYPE=WAIT,DCI=SINGLE macro call was issued. The task is waiting for a single ECB to be posted, after which the task may be resumed.

    - A DFHKC TYPE=WAIT,DCI=TERMINAL macro call was issued. CICS has suspended the task. The task is waiting for terminal I/O to complete, after which the task may be resumed.

  - The Miscellaneous Wait type is issued by Task Control when the task is suspended on a resource type of EKCWAIT after issuing an EXEC CICS WAIT EVENT command. Task Control waits tend to be application-dependent. You should review the "Task Control Waits" part of Section 2.3: Dealing with waits (Bookmanager document) of the CICS/ESA Version 4.1 Problem Determination Guide.

- **Temporary Storage Waits**. Temporary storage is a scratchpad facility that is heavily used in many systems. Temporary storage exists in either main storage above the 16MB line (ECDSA), or auxiliary storage in a VSAM-managed data set. Temporary storage waits are related to temporary storage existing in auxiliary storage.

  A task is forced to wait on temporary storage in auxiliary storage if the task has made an unconditional request for temporary storage, and the request cannot be met because insufficient auxiliary storage is available

  There are two likely reasons why a task might be suspended waiting for temporary storage:

- The task has issued a request requiring too large a piece of temporary storage.

- The task has issued a request requiring a reasonable amount of temporary storage, but there is too little available. This could indicate that the amount of auxiliary storage is becoming exhausted. Alternatively, there could be a relatively large amount of auxiliary storage available, but the storage is too fragmented for the request to be satisfied.

The Temporary Storage Statistics part of the CICS interval statistics contain information which can be analyzed to determine whether the WLM Miscellaneous Wait was likely caused by a Temporary Storage wait. The field A12STA8F field is a count of the number of times a task was suspended or had been abended because auxiliary storage had been exhausted.

- If the A12STA8F value is zero, you can be sure that the WLM Miscellaneous waits were **not** caused by Temporary Storage waits.

- If the A12STA8F value is non-zero, it is possible that the WLM Miscellaneous waits **were** caused by Temporary Storage waits. Unfortunately, there is no way to determine whether a task suspended for Temporary Storage constraint actually was in the service class missing its performance goal. However, tasks in the CICS region are encountering waits for Temporary Storage if the A12STA8F value is non-zero, and you should normally consider action. Further, the waiting task may be automatically purged[7] if it has waited for temporary storage longer than the deadlock time-out parameter specified in the installed transaction definition. Otherwise, it is not purged, and is liable to be suspended indefinitely.

  If the A12STA8F value is non-zero, CPExpert suggests that you review the suggested actions beginning on page 289 of the IBM *CICS Verson 4.1 Performance Guide*. These actions provide a checklist for improving the performance of temporary storage residing on auxiliary storage.

Alternatively, you can execute the CICS Component of CPExpert against the CICS region(s) serving the service class missing its performance goal. The CICS Component will analyze the CICS interval statistics to identify performance problems.

[7]Certain conditions prevent purging of a task (as examples, a deadlock time-out value of 0, or a specification of SPURGE(NO)).

- **Transient Data waits**. Tasks issuing requests to read and write to transient data destinations can be suspended for several reasons. The reasons depend on the type of request being made, and whether the task is attempting to access an extrapartition or an intrapartition queue. One of the reasons a task is suspended is related to the TD_INIT resource type, and occurs during system initialization.

  A second stage PLT program being executed during system initialization can issue a request for a resource that is not yet available, because the component that services the request has not yet been initialized. If the program issues a transient data request that cannot yet be serviced, it is suspended on a resource type of TD_INIT with a resource name of DCT. CICS provides the Workload Manager with a Miscellaneous Wait when a task is waiting on the TD_INIT resource type.

  Workload Manager Miscellaneous Waits for Transient Data occur only during system initialization. These waits would not cause a service class to miss its performance goal because the region has not yet begun accepting transactions.

- **User waits**. CICS provides the Workload Manager with a Miscellaneous Wait when a task is waiting on an ECB list posted by the user. User waits are application dependent.

- **VTAM waits**. CICS provides the Workload Manager with a Miscellaneous Wait when a task is waiting on three resource types: ZCIOWAIT, ZCZGET, and ZCZNAC.

  - The ZCIOWAIT resource type wait is caused by a task waiting on terminal I/O.

  - The ZCZGET resource type wait is caused with application request logic for LU6.2 devices.

  - The ZCZNAC resource type wait is for DFHZNAC to issue an error message.

- **XRF alternate system waits**. CICS provides the Workload Manager with a Miscellaneous Wait when a task is waiting caused by XRF alternative system waits. The XRF takeover process is a major system event, and you would not expect individual tasks to perform well during the takeover.

To summarize the above discussion, the most likely causes of Workload Manager Miscellaneous Waits, during **normal** transaction processing, are:

(1) CICS system task waits, (2) storage waits, (3) temporary storage waits, and (4) application-dependent waits.

- You should consider placing CICS system tasks into a single service class. IBM suggests that you not mix CICS-supplied transactions in a service class with user transactions. Once this has been done, remaining waits are likely to be related to SUSPENDED user tasks.

- You can examine CICS interval statistics to determine whether the Miscellaneous Waits are related to storage waits or temporary storage waits. The preceeding discussion describes the relevant fields in the interval statistics. Alternatively, you can execute the CICS Component of CPExpert against the CICS region(s) serving the service class missing its performance goal. The CICS Component will analyze the CICS interval statistics to identify performance problems.

- If you have taken the above actions and Miscellaneous Waits remain a major cause of transaction delay during normal operations, the most likely cause is application-dependent waits. You may wish to examine applications to determine whether they cause the waits, or you may simply ignore the waits.

Reference: CICS/ESA Version 4.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section

CICS/ESA Version 4.1 Problem Determination Guide)
Section 2.3: Dealing with waits

CICS/ESA Front End Programming Interface User Guide)
Section 2.4.2 (Performance) - system-related performance
Section 3.4.5.2 (Performance) - application-related performance

CICS/TS Release 1.1 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section

CICS/TS Release 1.1 Problem Determination Guide)
Section 2.3: Dealing with waits

CICS/TS Release 1.2 Performance Guide
Section 2.7.1.1: The response time breakdown in percentage section

CICS/TS Release 1.2 Problem Determination Guide)
Section 2.3: Dealing with waits

©Copyright 1994, Computer Management Sciences, Inc.

CICS/TS Release 1.3 Performance Guide
Section 2.6.1.1: The response time breakdown in percentage section

CICS/TS Release 1.3 Problem Determination Guide)
Section 2.3: Dealing with waits

CICS/TS Front End Programming Interface User Guide
Section 2.4.2 (Performance) - system-related performance
Section 3.4.5.2 (Performance) - application-related performance

CICS/TS for z/OS Release 2.2 *Performance Guide*:
Chapter 8: Understanding RMF Workload Manager Data

CICS/TS for z/OS Release 2.2 Problem Determination Guide)
Chapter 6: Dealing with waits

CICS/TS for z/OS Release 2.2 Front End Programming Interface User Guide
Chapter 6: FEPI Performance
Chapter 14: Application Design (Performance)

Thanks: Computer Management Sciences would like to recognize the efforts of the IBM CICS/ESA Development Team, IBM United Kingdom Laboratories (particularly Mr. Chris Baker) for providing detailed information about the resources that a CICS task might be waiting on. Based on an informal request to Chris at the August 1995 SHARE Technical Conference, IBM revised its CICS Problem Determination Guide (see above reference) to include a detailed itemization of the CICS waits. This invaluable information allows CPExpert to provide a more comprehensive analysis of CICS delays.

Rule WLM133: Significant transaction time was switched in sysplex

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent switched to another system in the sysplex. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent switched to another system in the sysplex.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1996, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Switched state indicates that processing of the transaction had been switched from the work manager (e.g., a CICS region) that was providing information to the Workload Manager. The transaction could have been switched to another CICS region (for example) in the same MVS image, switched to another MVS image in the sysplex, or switched to somewhere in the network.

- **Switched in the MVS image**. When the transaction is switched to another subsystem in the same MVS image, the subsystem from which the transaction is being shipped indicates that the monitoring environment transaction is being transferred to another subsystem (another "server). The receiving subsystem provides transaction delay information to the Workload Manager.

  CPExpert will acquire information about the server service class to which the transaction is switched. The server information will be analyzed to identify delays. If the server serves multiple transaction service classes, CPExpert prorates the delays based on amount of service provided to the different transaction service classes (the service information is contained in the R723SCS# variable in SMF TYPE 72 records). Other rules

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

provide information about delays when a transaction has been switched in the MVS image (for example, Rule WLM120 to Rule WLM132 provide information about the transaction delays. Rules WLM150-WLM152, WLM210, WLM211, etc. provide information about the server executing in the same MVS image.)

- **Switched in the sysplex**. When the transaction is switched to or switched to another MVS image in the sysplex, the subsystem from which the transaction is being shipped indicates that the monitoring environment transaction is being transferred to another subsystem. The receiving subsystem on the new MVS image provides transaction delay information to the Workload Manager.

  CPExpert will acquire information about the server service class to which the transaction is switched. The server information will be analyzed to identify delays.

  One unfortunate aspect of the information is that there is no way to relate delays to a server with the system on which the transaction originated.

  For example, a CICSRGN server service class on SYSA might provide service to several transaction service classes, both those originating on SYSA and those shipped from a number of other MVS images.

  There is no way to relate the delays in CICSRGN with the transaction service classes and the MVS images on which they originate.

  CPExpert provides Rule WLM133 when a significant amount of transaction delay can be attributed to the "switched in the sysplex" state. Rule WLM133 is provided to alert you to the possibility that the server analysis is flawed.

- **Switched in the network**. If the transaction is switched somewhere in the network. the Workload Manager has no more information about the status of the transaction; it is simply "switched in the network" from the Workload Manager's view.

  CPExpert provides Rule WLM134 when a significant amount of transaction delay can be attributed to the "switched in the sysplex" state. Rule WLM134 is provided to explain why further analysis is not possible.

CPExpert produces Rule WLM133 when the primary or secondary cause of delay was that the transaction service class was in the Switched in the Sysplex state for a significant percent of its response time.

The following example illustrates the output from Rule WLM105 (to show the primary cause of delay), followed by the output from Rule WLM133:

```
RULE WLM105:  SERVICE CLASS DID NOT ACHIEVE PERCENTILE RESPONSE GOAL

   Service Class CICSPROD did not achieve its response goal during the
   measurement intervals shown below.  The response goal was 90.00 percent
   of the transactions completing within 1.000 seconds, with an importance
   level of 3.  CICSPROD was defined as a "served" Service Class (e.g.,
   IMS or CICS transactions).  The below causes of delay were based upon
   BEGIN_TO_END PHASE samples.  CICSPROD was served by CICSRGN.

                                     TRANS      %
                             TOTAL  MEETING  MEETING PERF  PRIMARY,SECONDARY
   MEASUREMENT INTERVAL      TRANS    GOAL     GOAL  INDX  CAUSES OF DELAY
   10:00-10:30,26MAR1996     6,849   5,383     78.6  4.00  SYSPLEX(87%)
   10:30-11:00,26MAR1996     6,614   4,606     69.6  4.00  SYSPLEX(86%)
   11:00-11:30,26MAR1996     6,579   4,445     67.6  4.00  SYSPLEX(85%)
   11:30-12:00,26MAR1996     6,770   5,126     75.7  4.00  SYSPLEX(86%)
   12:30-13:00,26MAR1996     6,611   5,220     79.0  4.00  SYSPLEX(86%)
   13:00-13:30,26MAR1996     6,752   4,993     73.9  4.00  SYSPLEX(86%)


RULE WLM133:  SIGNIFICANT TRANSACTION TIME WAS SWITCHED IN SYSPLEX

   A significant amount of the transaction response time for the CICSPROD
   Service Class was spent switched to another MVS image in the sysplex.
   Please refer to the description of Rule WLM133 for a discussion of
   the implications of this finding on the analysis being done by CPExpert.
```

At present, there is little information provided regarding delays to transaction service classes once the transaction has been switched to another system. There exists at least the following possible delays:

- Queue delay in the system being analyzed (MRO/XCF delays or ISC delays caused by the system or by CICS parameters). These delays might be revealed by the CICS Component of CPExpert as it analyzes CICS performance constraints.

- Coupling facility delays. CPExpert will automatically analyze coupling facility statistics when Rule WLM133 is produced. This analysis may reveal problems with the coupling facility parameters.

- Delays in the system to which the transaction is being shipped. CPExpert will automatically analyze delays in all systems in which the transaction service class executes. There are several scenarios that complicate the analysis:

  - The sysplex is set up in a "standard" way in which a CICS Terminal Owning Region (TOR) is started in one system and CICSplex/SM is used switch transactions to Application Owning Regions (AORs) on a number of systems.

This is the simplest to evaluate, as there is some correlation between BTE Phase in the TOR system and Execution Phase time in the other systems. In this situation, the analysis by CPExpert is plausible.

- The sysplex is set up with TORs on more than one system, transactions can be submitted to the different TORs on different systems, and the transactions are switched among systems on the sysplex.

  It becomes unclear which system actually processes the transactions of a transaction service class missing its performance goal. (That is, the transactions might process satisfactorily on one system but not perform well on another system.)

  Further, depending on the transaction mix on different systems, there may be different delays to transactions on the different systems. It is entirely possible that performance may be acceptable on several systems, while performance is poor on one or more other systems.

  The analysis in this situation is suspect, at present. Perhaps as the CPExpert algorithms improve (or more data is available), the analysis will be more robust.

Suggestion: There are no suggestions with this finding, since it simply explains why CPExpert may not be able to provide meaningful information about the causes of delay for the service class missing its service goal on the system in which the service class delay was detected.

CPExpert will analyze the delays on each MVS image in which the transaction service class executed. Other rules will be produced to provide more information.

©Copyright 1996, Computer Management Sciences, Inc.

Rule WLM134: Significant transaction time was switched in network

Finding: A significant amount of the transaction response time for the service class missing its performance goal was spent switched outside the sysplex somewhere in the network. This finding applies to service classes which are part of a subsystem (e.g., CICS transactions).

Impact: This finding has MEDIUM IMPACT or HIGH IMPACT on performance of the service class. The level of impact depends on the percent of transaction response time spent switched to another system in the sysplex.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

CICS/ESA Version 4.1 reports two separate views of the transactions: the *begin_to_end phase* and the *execution phase*[2].

- **Begin_to_end phase**. The begin_to_end phase starts when CICS/ESA Version 4.1 has classified the transaction[3]. This action normally is done in a CICS Terminal Owning Region (TOR).

- **Execution phase**. The execution phase starts when either CICS/ESA Version 4.1 or IMS Version 5 has started an application task to process

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]IMS Version 5 reports only *execution phase* samples.

[3]Classifying the transaction into a service class is actually done by the Workload Manager when CICS issues the IWMCLSFY macro. Please refer to Section 4 for a more complete discussion of the subsystem work manager (e.g., CICS) interaction with the Workload Manager.

©Copyright 1996, Computer Management Sciences, Inc.

the transaction. For CICS, this normally is done in a CICS Application Owning Region (AOR).

Within each phase, CICS or IMS reports the "state" of the transaction, from the view of CICS or IMS. The state of the transaction is reported in the following categories[4]:

- **Idle state**.
- **Active state**.
- **Ready state**.
- **Wait state**.
- **Switched state**.

If the subsystem supports work manager delay reporting, the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. When a transaction service class fails to achieve its performance goal, CPExpert analyzes the information to identify the primary and secondary causes of delay.

The Switched state indicates that processing of the transaction had been switched from the work manager (e.g., a CICS region) that was providing information to the Workload Manager. The transaction could have been switched to another CICS region (for example) in the same MVS image, switched to another MVS image in the sysplex, or switched to somewhere in the network.

- **Switched in the MVS image**. When the transaction is switched to another subsystem in the same MVS image, the subsystem from which the transaction is being shipped indicates that the monitoring environment transaction is being transferred to another subsystem (another "server). The receiving subsystem provides transaction delay information to the Workload Manager.

  CPExpert will acquire information about the server service class to which the transaction is switched. The server information will be analyzed to identify delays. If the server serves multiple transaction service classes, CPExpert prorates the delays based on amount of service provided to the different transaction service classes (the service information is contained in the R723SCS# variable in SMF TYPE 72 records). Other rules

[4]Please refer to Section 4 of this document for a more comprehensive discussion of the transaction states and the interaction between the subsystem (CICS or IMS) and the Workload Manager.

provide information about delays when a transaction has been switched in the MVS image (for example, Rule WLM120 to Rule WLM132 provide information about the transaction delays. Rules WLM150-WLM152, WLM210, WLM211, etc. provide information about the server executing in the same MVS image.)

- **Switched in the sysplex**. When the transaction is switched to or switched to another MVS image in the sysplex, the subsystem from which the transaction is being shipped indicates that the monitoring environment transaction is being transferred to another subsystem. The receiving subsystem on the new MVS image provides transaction delay information to the Workload Manager.

  CPExpert provides Rule WLM133 when a significant amount of transaction delay can be attributed to the "switched in the sysplex" state.

- **Switched in the network**. If the transaction is switched somewhere in the network. the Workload Manager has no more information about the status of the transaction; it is simply "switched in the network" from the Workload Manager's view.

  CPExpert provides Rule WLM134 when a significant amount of transaction delay can be attributed to the "switched in the sysplex" state.

The following example illustrates the output from Rule WLM105 (to show the primary cause of delay), followed by the output from Rule WLM134:

```
RULE WLM134:  SIGNIFICANT TRANSACTION TIME WAS SWITCHED OUTSIDE SYSPLEX

  A significant amount of the transaction response time for the APPCGRPA
  Service Class was spent switched outside the sysplex, to somewhere in
  the network.  No additional information is available in SMF records, and
  no further analysis can be done.
```

Suggestion: There are no suggestions with this finding, since it simply explains why further analysis is not possible.

Rule WLM135: IMS activity processing transactions in service class

Finding: CPExpert has detected that a large percent of the transaction response time was related to IMS activity involved in processing the transactions in the service class.

Impact: This finding means that transactions were waiting for IMS activity - either an IMS Message Processing Region was processing the transaction or an IMS Message Processing Region was waiting for some reason.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: A transaction service class could be “served” by CICS regions, by IMS regions, by DB2 threads, or a combination of these. When a transaction service class misses its performance goal, CPExpert determines whether transaction delay information is available, from the view of these “server” subsystems.

When a transaction service class fails to achieve its performance goal, CPExpert analyzes the delay information to identify the primary and secondary causes of delay.

If the subsystem supports work manager delay reporting (that is, the subsystem is at CICS Version 4 or above, IMS Version 5 or above, or DB2 Version 6 or above), the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. Field R723RTYP describes the subsystem that reports the transaction delay information (e.g., CICS, IMS, DB2, etc.).

When a significant amount of transaction time is spent in IMS, CPExpert examines the delay information reported by IMS. This reports the result of that analysis.

IMS Version 5 reports only one view of the transactions: the *execution phase*. The execution phase starts when IMS Version 5 has started an application task to process the transaction in a Message Processing Region (MPR). IMS does not report on the Begin_to_end Phase as does some subsystems (for example, CICS reports both Begin_to_end Phase

©Copyright 1997, Computer Management Sciences, Inc.

and Execution Phase). IMS designers apparently believed that so little time was spent in the Message Control Region that little benefit would be gained by reporting transaction states in the Message Control Region[1]. Consequently, only Execution Phase information is provided by IMS.

IMS reports the transaction states in the following categories within the Execution Phase:

- **Idle state**. The Idle state means that the IMS transaction is waiting for work.
- **Active state**. The Active state means that IMS is executing an application program on behalf of the transaction.
- **Waiting for I/O state**. The Waiting for I/O state means that IMS had initiated some I/O operation and is waiting for completion.
- **Waiting for Lock state**. The Waiting for Lock state means that IMS is waiting on a lock request.

CPExpert uses Rule WLM135 to report the time when a "served" transaction service class was served by IMS. The information is provided relative to the total subsystem samples reported by SMF for the transaction service class missing its goal. Thus, a CPExpert user can see the effect of IMS activity and waiting on the transaction response time.

The following example illustrates the output from Rule WLM135:

```
RULE WLM135: IMS ACTIVITY IN SUPPORT OF SERVICE CLASS

   CICSPROD: The following information shows the distribution of samples
   in IMS for those periods when IMS accounted for a significant part
   of the response time of the CICSPROD Service Class.  The percentages
   are shown relative to the total  samples for the CICSPROD Service
   Class.

                                 PCT  IMS      PCT IMS         PCT IMS
   MEASUREMENT INTERVAL           ACTIVE     WAIT FOR I/O   WAIT FOR LOCK
   13:00-13:30,01MAR2001           42.9          0.0            0.0
```

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules may be produced to provide more information. Please refer to Rule WLM104 or Rule WLM105 for information about the causes of delay to the subsystem transaction service classes.

[1] This is an interesting belief, since the IMS Administration Guide specifically states that a major part of transaction delay time in a busy system could be caused by delays in the IMS Control Region.

Reference: IMS/ESA V5 Administrative Guide: System

Section 6.1.2.6: Interpreting MVS WLM Change State PB Service Codes
Section 6.5: Transaction Flow

IMS/ESA V6 Administrative Guide: System

Section 2.2.1.2.6: Interpreting MVS WLM Change State PB Service Codes
Section 2.2.5: Transaction Flow

IMS/ESA V7 Administrative Guide: System

Section 2.2.1.2.6: Interpreting MVS WLM Change State PB Service Codes
Section 2.2.5: Transaction Flow

©Copyright 1997, Computer Management Sciences, Inc.

Rule WLM136: DB2 activity processing transactions in service class

Finding: CPExpert has detected that a large percent of the transaction response time was related to DB2 activity involved in processing the transactions in the service class.

Impact: This finding means that transactions were waiting for DB2 thread activity.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: A transaction service class could be “served” by CICS regions, by IMS regions, by DB2 threads, or a combination of these. When a transaction service class misses its performance goal, CPExpert determines whether transaction delay information is available, from the view of these “server” subsystems.

When a transaction service class fails to achieve its performance goal, CPExpert analyzes the delay information to identify the primary and secondary causes of delay.

If the subsystem supports work manager delay reporting (that is, the subsystem is at CICS Version 4 or above, IMS Version 5 or above, or DB2 Version 6 or above), the delay information is available in the "Work Manager/Resource Manger State Section" of SMF Type 72 (Subtype 3) records. Field R723RTYP describes the subsystem that reports the transaction delay information (e.g., CICS, IMS, DB2, etc.).

With Version 6, DB2 uses the execution delay monitor services provided by the Workload Manager. These services are used to inform the Workload Manager about DB2's view of the current state of a work request or thread, such as *ready for execution* (active) or *waiting for execution* (suspended).

When a significant amount of transaction time is spent in DB2 (that is, R723RTYP = ‘DB2’), CPExpert examines the delay information reported by DB2. This Rule (Rule WLM136) reports the result of that analysis.

DB2 Version 6 reports transaction states in the following categories:

©Copyright 1997, Computer Management Sciences, Inc.

- **Active state**. The Active state means that the DB2 thread is ready for execution. Although the thread is marked as Active, a thread may be active only from DB2's point of view. The thread actually might be delayed due to a page fault, for CPU access, etc.

- **Waiting for I/O state**. The Waiting for I/O state means that DB2 had initiated some I/O operation and the thread was suspended waiting for I/O completion.

- **Waiting for Lock state**. The Waiting for Lock state means that the DB2 thread is suspended while DB2 is acquiring a lock.

- **Waiting for New Latch state**. The Waiting for Latch state means that the DB2 thread is suspended while DB2 is acquiring a latch.

- **Waiting for Network Delay state**. The Waiting for Network Delay state means that the DB2 thread is suspended while DB2 is waiting for a session to be established somewhere in the network.

- **Waiting for Miscellaneous Reasons state**. The Waiting for Miscellaneous Reasons state normally means that the work manager could not readily identify the cause of the waiting. With DB2 threads, this state often means that the DB2 thread is suspended waiting for a stored procedure to be scheduled (queuing for stored procedure).

CPExpert uses Rule WLM136 to report the time when a “served” transaction service class was served by DB2. The information is provided relative to the total subsystem samples for the transaction service class missing its goal. Thus, a CPExpert user can see the effect that DB2 activity and waiting has on the transaction response time.

The following example illustrates the output from Rule WLM136:

```
RULE WLM136: DB2 ACTIVITY IN SUPPORT OF SERVICE CLASS

   TENTHSEC: The following information shows the distribution of samples
   in DB2 for those periods when DB2 accounted for a significant part
   of the response time of the TENTHSEC Service Class.  The percentages
   are relative to the total samples for the TENTHSEC Service Class.

                        PCT DB2  --PERCENT OF SAMPLES DB2 WAS WAITING---
   MEASUREMENT INTERVAL  ACTIVE  I/O  LOCK  LOCSES  PLXSES  NETSES  MISC
   13:29-13:44,14MAR2001  14.8  23.5   0.0    0.0     0.0     0.0    7.8
   13:44-13:59,14MAR2001   8.2   6.6   0.0    0.0     0.0     0.0    6.6
   13:59-14:14,14MAR2001   3.5   2.6   0.0    0.0     0.0     0.0    1.7
```

Suggestion: There are no suggestions with this finding. CPExpert will continue analysis and other rules may be produced to provide more information. Please refer to Rule WLM104 or Rule WLM105 for information about the causes of delay to the subsystem transaction service classes.

Reference: DB2 UDB for OS/390 Version 6 Performance Topics Redbook (SG24-5351-00)

Rule WLM140: Sysplex performance index was significantly less than local performance index

Finding: The average sysplex performance index was significantly less than the average performance index on the local system. This finding applies only to environments that have multiple systems in the sysplex running under Goal Mode.

Impact: This finding can have a HIGH IMPACT on performance of the service class period.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: As described in Section 4 (Chapter 3.5: Policy Adjustment), the Workload Manager periodically assesses the performance of each service class period, comparing the performance achieved by the service class period against the performance goals specified for the service class period. The comparison of performance is based on the performance index computed for the service class periods, and on the goal importance of the service class periods.

The Workload Manager initially assesses performance based on the *sysplex performance index* computed for each service class period. This assessment is done at each goal importance level. Policy adjustment actions are evaluated for the worst-performing service class period at the highest goal importance, then the next worst-performing, etc. It is important to realize that only one service class period will be "helped" by the policy adjustment algorithms per policy adjustment interval[1].

If the Workload Manager has evaluated the performance of all service class periods at the highest goal importance *based on sysplex performance*

[1]Recall that the policy adjustment interval is 10 seconds of elapsed time.

©Copyright 1997, Computer Management Sciences, Inc.

index and no action has been taken, the next step depends on whether APAR OW25542 has been applied[2].

- **OW25542 has not been applied**. With the normal logic, the Workload Manager will examine the performance of all service class periods at the next-highest goal importance *based on sysplex performance index*. The Workload Manager will continue analyzing performance at successively lower goal importance levels, based on sysplex performance index. After the performance of all service class periods with goals have been analyzed with no action, the Workload Manager will perform the analysis beginning with the highest goal importance, **using the local performance index** as the measure of performance,.

- **OW25542 has been applied**. With OW25542, the Workload Manager will examine the performance of all service class periods at the highest goal importance **using the local performance index** as the measure of performance. The Workload Manager will continue examining performance at successively lower goal importance levels, analyzing performance based on sysplex performance index followed by an analysis of performance based on local performance index.

Both the original design of the Workload Manager and the fix for OW25542 operate under a basic assumption: that a sysplex consists of multiple systems configured in a symmetric manner, and that service class periods can operate on any system in the sysplex. If the workload being processed consists of transaction service classes such as CICS transactions managed by CICSplex/SM and routed to any system in the sysplex to be processed in cloned CICS regions, this view of the sysplex makes sense.

From this perspective, all systems in the sysplex can be viewed collectively as a pool of resources and the performance of the transactions can be evaluated based on how well the transactions perform on the sysplex. If a service class period is not meeting its performance goal *on the sysplex*, action may or may not be necessary at a local system level. Consequently, **sysplex performance index** is the basic measure of performance used in the Workload Manager design.

Unfortunately, this logic does not work in all situations. Consider a site that has established a service class for TSO trivial transactions. The TSO users might log onto, for example, two systems: SYSA and SYST. The users on SYSA might represent production work while the users on SYST might represent TSO testing (and might not be as important to the site as the production work).

[2]The fix for OW25542 is scheduled for release in May, but may not be available for all versions of MVS at that date. OW25542 describes Importance 1 work (the highest importance), while the problem actually applies to any importance level.

It is conceivable that the test TSO user could receive good response while some of the production TSO users could receive very poor response. From a sysplex performance index calculation, the test and production response times would be grouped together by the algorithm. Depending on the distribution of response times, the sysplex performance index might be relatively low.

One result of this could be that the Workload Manager would not attempt to "help" the production TSO service class since the sysplex performance index might indicate that there was no performance problem. However, the production users might feel quite differently about the performance!

CPExpert evaluates performance based on a calculated average **local performance index** for each service class period. This is because we believe that the Workload Manager approach is fatally flawed in practically every existing environment. There **will** be environments with the sysplex-centric view will be a proper way to evaluate performance, but few such environments exist today. Rather, most environments operating in Goal Mode run in a monoplex, or in a sysplex with a wide variety of work executing on different systems.

Consequently, CPExpert evaluates performance at the local system level, and makes suggestions or comments based on potential performance improvement actions at the local system level.

On the other hand, the Workload Manager does evaluate the sysplex performance index as the primary indicator of performance. Thus, CPExpert computes the average sysplex performance index and displays both the local performance index and sysplex performance index in appropriate rules.

When CPExpert detects that a service class period misses its performance goal (based on the local performance index), CPExpert examines the sysplex performance index. If the sysplex performance index is *significantly less* than the local performance index, the Workload Manager might take no action to improve performance for the service class. CPExpert reports this potential problem via Rule WLM140. Rule WLM140 is produced when the sysplex performance index is less than 75% of the local performance index.

The following example illustrates the output from Rule WLM140:

```
RULE WLM140: SYSPLEX PERFORMANCE INDEX WAS SIGNIFICANTLY LESS THAN LOCAL

   IMS (Period 1):  The sysplex performance index for this service class
   period was significantly less than the local performance index.  One
   implication of this is that the Workload Manager might not attempt to
   improve performance of the service class period on the local system.
   Please refer to the WLM Component User Manual for a discussion of how
   the sysplex performance index and local performance index are used by
   the Workload Manager.  This finding applies to the following measurement
   intervals:

                                   PERFORMANCE INDEX
   MEASUREMENT INTERVAL            LOCAL     SYSPLEX
   11:00-11:15,06MAR1997            1.83       0.97
   11:15-11:39,06MAR1997            2.14       0.82
```

Suggestion: If this finding occurs, CPExpert suggests that you review the relative values of the sysplex performance index and the local performance index presented by Rule WLM140.

You should be concerned if the sysplex performance index is significantly less than the local performance index for important work since this would indicate that the Workload Manager **might not** take action to improve performance on the local system.

You should become alarmed if the sysplex is less than 1.0 for important work, since this would indicate that the Workload Manager probably **would not** take action to improve performance on the local system!

In either case, you should consider the following alternatives:

- Review the information presented with the predecessor rules and other rules related to the "missed goal" analysis for the service class period. Based on this review and considering the importance of the work in the service class period, you should assess whether any action is necessary or whether you should ignore the finding. If you ignore the finding, you should be aware that the Workload Manager might not take actions to improve the performance of the service class period.

- If you decide that action is warranted, you should revise the workload classification scheme to place the work assigned to the service class period missing its goal into a different service class. This might involve creating a new service class for the work executing on the local system, or creating a new service class for the work executing elsewhere in the sysplex.

Reference: MVS Programming: Workload Management Services

MVS/ESA(SP 5):	Chapter 4: Using SMF Record Type 99
OS/390 (V1R1):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R2):	Chapter 7: Using SMF Record Type 99
OS/390 (V1R3):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R4):	Chapter 9: Using SMF Record Type 99
OS/390 (V2R5):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R6):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R7):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R8):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R9):	Chapter 10: Using SMF Record Type 99
OS/390 (V2R10):	Chapter 10: Using SMF Record Type 99
z/OS (V1R1):	Chapter 10: Using SMF Record Type 99
z/OS (V1R2):	Chapter 10: Using SMF Record Type 99
z/OS (V1R3):	Chapter 10: Using SMF Record Type 99
z/OS (V1R4):	Chapter 10: Using SMF Record Type 99

©Copyright 1997, Computer Management Sciences, Inc.

Rule WLM150: Server service class delays (single transaction service class)

Finding: CPExpert has identified delays for the server service class which provided service to a subsystem transaction service class.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104: Subsystem Service Class did not achieve average response goal

Rule WLM105: Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Please refer to Rule WLM120 to Rule WLM132 for a discussion of the delays from the served service class.

After analyzing the **served** service class delays, CPExpert identifies the **server** service class. The server service class normally will be one or more CICS regions or IMS regions.

CPExpert analyzes the following possible delays to response time[1]:

- **CPU Using delay**
- **Denied CPU delay**
- **CPU Capping delay**
- **Swap-in delay**
- **MPL delay**
- **Page-in delay**
- **I/O delay**

[1]Please see Section 4 (Chapter 3.3) for a description of these delays.

©Copyright 1994, Computer Management Sciences, Inc.

- **Unknown delay**

CPExpert produces Rule WLM150 to provide a summary of the delay for the server service class.

The output from Rule WLM150 does not contain the MPL delay or swap-in delay. In most environments, server service classes are non-swappable and the MPL delay and swap-in delay columns would always show zero. Consequently, CPExpert does not clutter up the output with columns which almost always would be zero. However, CPExpert **does** analyze these delays if any are non-zero.

More than one server service class might serve the subsystem transaction service class which missed its performance goal. In this case, CPExpert produces multiple Rule WLM150 findings - one for each server service class. CPExpert then analyzes the delays for each server service class.

The following example illustrates the output from Rule WLM150:

```
RULE WLM150:  SERVER SERVICE CLASS DELAYS

   The IMS Service Class was served by the IMSCTL Service Class.
   The IMSCTL Service Class experienced the following delays during the
   measurement intervals when the IMS Service Class missed its
   performance goal (the delays are shown relative to the active time
   of IMSCTL):
                                                                PCT      PCT
                              PCT CPU   PCT CPU   PCT CPU   PAGING  UNKNOWN
   MEASUREMENT INTERVAL        USING    DELAYED   CAPPING    WAIT     WAIT
   13:07-13:12,21JUN1994        13.3      86.7      0.0       0.0      0.0


RULE WLM150:  SERVER SERVICE CLASS DELAYS

   The IMS Service Class also was served by the IMSMP Service Class.
   The IMSMP Service Class experienced the following delays during the
   measurement intervals when the IMS Service Class missed its
   performance goal (the delays are shown relative to the active time
   of IMSMP):
                                                                PCT      PCT
                              PCT CPU   PCT CPU   PCT CPU   PAGING  UNKNOWN
   MEASUREMENT INTERVAL        USING    DELAYED   CAPPING    WAIT     WAIT
   13:07-13:12,21JUN1994         0.5      24.4      0.0       0.0     75.1
```

In the above example, IMS transactions were placed in the IMS Service Class. The IMS Service Class was served by an IMS control region (the IMSCTL Service Class) and IMS message processing region (the IMSMP Service Class). Rule WLM150 is produced for both servers, to show the delays to the servers.

The delay information is shown relative to the active time of the server service class, and the percentages will total 100%.

There is no information in SMF Type 72 records which shows how much of the response time of the **served** service class (e.g., the IMS Service Class) could be attributed to delays in the **individual servers** (e.g., IMSCTL or IMSMP).

If the individual servers serve more than one service class, there is information in the SMF Type 72 records to show how many **times** an address space in the server was observed to be providing service to the served service class. In this example, the IMSMP Service Class could have served (1) the IMS Service Class and (2) several other service classes[2]. SMF data would show how many times the IMSMP Service Class provided service to IMS, IMS1, IMS2, etc.

The WLM counts each time the server issues the IWMRPT macro to indicate that a transaction has completed. This count lets the WLM know how many times the server (e.g., a CICS region) provided service to the served service class (e.g., CICS transactions).

Additionally, every 250 milliseconds, the WLM samples server service classes to see which served service classes they are serving. The sampling process ensures that the WLM keeps track of service provided to long-running transactions.

SMF field R723SCS# contains a summary of the count and samples[3]. This field can be used to apportion the service provided by the server to the various transaction service classes being served.

Suggestion: There are no suggestions directly associated with this finding. CPExpert will continue analysis of the server service class(es), and other rules should be produced to provide suggestions.

[2]For example, IMS1 Service Class, IMS2 Service Class, etc.

[3]SMF field R723SCS# was improperly described in early versions of the SMF manual. The field has been modified to conform with the above description after CPExpert advised IBM of the error.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM151: Server service class delays (multiple transaction service classes)

Finding: CPExpert has identified delays for the server service class which provided service to a subsystem transaction service class. The server provided service to more than one subsystem transaction service class, and CPExpert prorates the service provided to the different transaction service classes.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Please refer to Rule WLM120 to Rule WLM132 for a discussion of the delays from the served service class.

After analyzing the **served** service class delays, CPExpert identifies the **server** service class. The server service class normally will be one or more CICS regions or IMS regions.

CPExpert analyzes the following possible delays to response time[1]:

- **CPU Using delay**
- **Denied CPU delay**
- **CPU Capping delay**
- **Swap-in delay**
- **MPL delay**

[1]Please see Section 4 (Chapter 3.3) for a description of these delays.

©Copyright 1996, Computer Management Sciences, Inc.

- **Page-in delay**

- **I/O delay**

- **Unknown delay**

If the server service class provides service to more than one transaction service class, CPExpert must apportion the resource utilization and delays to the different transaction service classes. The apportioning is done based on the value of SMF variable R723SCS# for each transaction service class.

CPExpert produces Rule WLM151 to provide a summary of the delay for the server service class.

The output from Rule WLM151 does not contain the MPL delay or swap-in delay. In most environments, server service classes are non-swappable and the MPL delay and swap-in delay columns would always show zero. Consequently, CPExpert does not clutter up the output with columns which almost always would be zero. However, CPExpert **does** analyze these delays if any are non-zero.

CPExpert produces Rule WLM151 to show the delays to the server service class. More than one server service class might serve the subsystem transaction service class which missed its performance goal. In this case, CPExpert produces multiple Rule WLM151 findings - one for each server service class. CPExpert then analyzes the delays for each server service class.

Additionally, a server service class might serve more than one transaction subsystem service class (in fact, this is the more common case). For example, a CICS region often will serve several transaction service classes composed of CICS transactions. In this case, CPExpert must apportion the resources used and delays encountered among the transaction service classes being served. The resources and delays encountered are reported by Rule WLM151.

Rule WLM151 also shows the percent of service provided the transaction service class missing its performance goal, relative to the service provided by the server to all transaction service classes.

- There is no information in SMF Type 72 records which shows how much of the response time of the **served** service class (e.g., the IMS Service Class) could be attributed to delays in the **individual servers** (e.g., CICSRGN).

- If the individual servers serve more than one service class, there is information in the SMF Type 72 records to show how many **times** an address space in the server was observed to be providing service to the served service class.

  The WLM counts each time the server issues the IWMRPT macro to indicate that a transaction has completed. This count lets the WLM know how many times the server (e.g., a CICS region) provided service to the served service class (e.g., CICS transactions).

  Additionally, every 250 milliseconds, the WLM samples server service classes to see which served service classes they are serving. The sampling process ensures that the WLM keeps track of service provided to long-running transactions.

SMF field R723SCS# contains a summary of the count and samples[2]. This field can be used to apportion the service provided by the server to the various transaction service classes being served.

The following example illustrates the output from Rule WLM151:

```
RULE WLM151:  SERVER SERVICE CLASS DELAYS

   The CICUSRTX Service Class was served by the CICSRGN Service Class.
   The CICSRGN Service Class experienced the following delays during the
   measurement intervals when the CICUSRTX Service Class missed its
   performance goal (the delays are shown relative to the EXECUTING time
   of CICSRGN).  CICSRGN also served other service classes.  The "PCT SERVED"
   column reflects the percent of service provided by CICSRGN to CICUSRTX,
   relative to the total service provided by CICSRGN to all service classes
   served by CICSRGN.

                                                          PCT       PCT
                          PCT CPU  PCT CPU  PCT CPU  PAGING  UNKNOWN  PCT
   MEASUREMENT INTERVAL    USING   DELAYED  CAPPING   WAIT     WAIT  SERVED
   13:07-13:12,21JUN1994   39.6     60.4     0.0      0.0      0.0   99.5
   13:17-13:22,21JUN1994   42.3     57.7     0.0      0.0      0.0   99.7
```

Suggestion: There are no suggestions directly associated with this finding. CPExpert will continue analysis of the server service class(es), and other rules should be produced to provide suggestions.

[2]SMF field R723SCS# was improperly described in early versions of the SMF manual. The field has been modified to conform with the above description after CPExpert advised IBM of the error.

Rule WLM152: Server served multiple transaction service classes

Finding: The server service class providing service to the transaction service class being analyzed by CPExpert provided service to transaction service classes other than the transaction service class missing its performance goal.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM151:	Server service class delays

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Please refer to Rule WLM120 to Rule WLM132 for a discussion of the delays from the served service class.

After analyzing the **served** service class delays, CPExpert identifies the **server** service class. The server service class normally will be one or more CICS regions or IMS regions.

If the server service class provides service to more than one transaction service class, CPExpert reports information about **all** transaction service classes served by the server. This information is provided for the RMF measurement intervals in which the transaction service class identified by Rule WLM151 missed its performance goal.

The following example illustrates the output from Rule WLM152:

©Copyright 1996, Computer Management Sciences, Inc.

```
RULE WLM152: SERVER SERVED MULTIPLE TRANSACTION SERVICE CLASSES

   Service Class CICSRGN served multiple transaction service classes during
   the intervals when CICUSRTX missed its performance goal.  Consequently,
   CPExpert must analyze the delays for each transaction service class
   separately, and must apportion the resources used by CICSRGN based on
   the number of times CICSRGN served each transaction service class.  The
   below information shows how often CICSRGN provided service to each
   transaction service class during intervals in which CICUSRTX missed its
   performance goal.

                              TRANSACTION       MISSED        PERCENT
   MEASUREMENT INTERVAL       SERVICE CLS       GOAL ?        SERVICE
   13:07-13:12,21JUN1994       CICUSRTX          YES            99.5
   13:07-13:12,21JUN1994       CICSYSTX          NO              0.5
   13:17-13:22,21JUN1994       CICUSRTX          YES            99.7
   13:17-13:22,21JUN1994       CICSYSTX          NO              0.3
```

Suggestion: CPExpert suggests that you review the information provided with Rule WLM152, to determine whether the distribution of service to the different transaction classes meets your installation objectives.

Rule WLM153: Server served multiple transaction service classes

Finding: The server service class providing service to the transaction service class being analyzed by CPExpert provided service to transaction service classes other than the transaction service class missing its performance goal.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM151:	Server service class delays

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Please refer to Rule WLM120 to Rule WLM132 for a discussion of the delays from the served service class.

After analyzing the **served** service class delays, CPExpert identifies the **server** service class. The server service class normally will be one or more CICS regions or IMS regions.

If the server service class provides service to more than one transaction service class, CPExpert reports information about **all** transaction service classes served by the server. This information is provided by Rule WLM152 for the RMF measurement intervals in which the transaction service class identified by Rule WLM151 missed its performance goal.

More than one transaction service class served by the same server could have missed its performance goal. Unnecessary and redundant output would be produced if CPExpert repeated Rule WLM152 for the same server, simply because another transaction service class missed its performance goal. Consequently, CPExpert produces Rule WLM153 to simply refer back to the earlier-produced Rule WLM152 if you wish to see which transaction service classes missed their performance goal.

©Copyright 1996, Computer Management Sciences, Inc.

The following example illustrates the output from Rule WLM153:

```
RULE WLM153: SERVER SERVED MULTIPLE TRANSACTION SERVICE CLASSES

   CICSRGN: Service class served multiple transaction service classes during
   the intervals when CICSCONV missed its performance goal.  Consequently,
   CPExpert must analyze the delays for each transaction service class
   separately, and must apportion the resources used by CICSRGN based on
   the number of times CICSRGN served each transaction service class.
   Please refer to the previous listing associated with Rule WLM152,
   which shows how often CICSRGN provided service to each transaction
   service class during intervals in which CICSCONV missed its performance
   goal.
```

Suggestion: CPExpert suggests that you review the information provided with Rule WLM152 produced earlier in the report, to determine whether the distribution of service to the different transaction classes meets your installation objectives.

Rule WLM170: Address spaces were idle a significant percent of time

Finding: The service class period being analyzed missed its response goal. However, address spaces in the service class were Idle for a significant percent of their overall active time. Consequently, the Workload Manager delay information may be meaningless.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101: Service Class did not achieve average response goal

Rule WLM102: Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM101 or Rule WLM102 to indicate that a service class did not achieve its response performance goal, the logic of these rules tries to identify the cause of the delay.

The Workload Manager periodically examines the SRM control blocks describing each address space and acquires samples[1] describing the state of each dispatchable unit of an address space (that is, each TCB or SRB associated with the address space). The Workload Manager accumulates the samples into counters which describe the state of the address space. The samples are summarized by service class period.

CPExpert analyzes the causes of delay to service class periods not meeting their response goal. Rule WLM101 and Rule WLM102 report the primary and secondary causes of delay to the response time.

For example, CPExpert might compute that the primary cause of delay to TSO transactions was that they were denied access to a processor for 35% of their active time, and that they were waiting for "unknown" causes[2] for another 30% of their active time.

CPExpert would report both these causes, and their respective percentages in Rule WLM102. CPExpert would continue analysis to

[1]With MVS/ESA SP5.1 Goal Mode, the sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

[2]Recall from Section 4 that the "unknown" cause is unknown as far as the System Resources Manager is concerned. The SRM identifies causes of delay only for those categories over which it has control. Delays over which the SRM has no control are grouped together into an "unknown" category. These delays typically are I/O delay, ENQ delay, waiting for cross-memory services, etc.

©Copyright 1996, Computer Management Sciences, Inc.

assess which service classes might deprive TSO transactions from access to a processor and to assess the likely causes of "unknown" delays.

The analysis performed by the Workload Manager and subsequent analysis by CPExpert is based on samples. The reliability of sampling depends upon having a sufficiently large number of samples such that the samples represent the "population" being sampled[3]. If a small number of samples are taken, invalid conclusions might be reached based on an analysis of the samples. In order for the conclusions about causes of delays to be valid, sufficient samples must be taken while address spaces were in a "ready" state rather than in an "idle" state.

When CPExpert determines that a service class with a response goal has missed its performance goal, CPExpert reviews the number of samples taken during times when address spaces were in a "ready" state. This number of samples is obtained by summing the CPU Using samples (R723CCUS), Total Delay samples (R723CTOT), and Unknown samples (R723CUNK). CPExpert produces Rule WLM171 if this total number of samples is small.

Once CPExpert has determined that an unacceptably small number of samples exist, no further analysis is done. It makes no sense to analyze delays to the service class based on a low number of samples, inasmuch as the conclusions from the samples would be invalid.

The following example illustrates the output from Rule WLM170:

```
RULE WLM170:  ADDRESS SPACES WERE IDLE A SIGNIFICANT PERCENT OF TIME

   The delay information presented above is based on the EXECUTION time of
   the TSOUSERS Server Class (the CPU Using, Execution Delay, and Unknown
   Delay).  These percentages show the distribution of time while some
   transaction was active.  However, address spaces in the TSOUSERS Service
   Class were IDLE for a significant percent of their overall active time.
   The below information shows the percent of total active time in which
   address spaces were executing (processing transactions) or were idle,
   and the average number of Workload Manager samples per transaction.
   Please refer to Rule WLM170 in the WLM Component User Manual for a
   discussion of the implications of this finding.

                          TOTAL     PCT      PCT  AVG SAMPLES  AVG SAMPLES
   MEASUREMENT INTERVAL   TRANS  EXECUTING  IDLE   PER TRANS   PER  MINUTE
   10:45-11:00,07DEC1994     63     4.0     96.0      2.3          9.5
   11:15-11:29,07DEC1994     32     3.0     97.0      3.1          7.0
   11:45-12:00,07DEC1994     14     1.2     98.8      3.1          2.9
```

Suggestion: CPExpert suggests that you consider the following alternatives:

[3]With the Workload Manager samples, the "population" consists of the possible execution states of address spaces being sampled.

- You can ignore the finding (and previous rules in the logic flow) if you feel that the situation is unusual rather than a continuing status. For example, the finding might be made when a service class was temporarily idle. For example, a TSO_SYS service class might be established for systems personnel to use only during certain times (e.g., a crisis situation). This service class might be idle for most of the time, but systems personnel might submit transactions periodically.

  If you chose to ignore the finding, you may wish to exclude the service class from analysis, using the EXCLUDE guidance parameters described in Section 2 (Chapter 1.1.8) of this document). You likely would become annoyed by CPExpert continually reporting that the service class missed its performance goal when you contemplate no action.

- If the service class reported by Rule WLM170 consists of Started Tasks, you should assess the important of the Started Tasks, and whether a response objective is proper. If the Started Tasks are important from a system view, you should consider allowing the Started Tasks to default to the SYSSTC service class. The SYSSTC service class has a high dispatching priority. Address spaces in SYSSTC will not be subject to the Workload Manager's dispatching priority adjustment algorithms[4].

- You may wish to delete the service class and assign the workload to a service class with more active address spaces if you feel that the situation is a continuing one. That is, if you feel that the address spaces normally are idle, you may wish to review whether they need their own service class. As general guidance, it is desirable to keep the service class periods to as small a number as possible.

[4]This alternative does not reduce the effect of the reduced preemption on address spaces in the service class. The alternative simply removes them from the Workload Manager's control.

Rule WLM171: Execution velocity was computed on a small sample set

Finding: The service class period being analyzed missed its execution velocity goal. However, the execution velocity was computed on a small sample set. Consequently, the execution velocity might be meaningless.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:
Rule WLM103: Service class did not achieve execution velocity goal

Discussion: Installations may specify an *execution velocity goal* for a service class period. An execution velocity is a measure of how fast work should run when the work is ready to run, without being delayed waiting for access to a CPU or delayed waiting for access to processor storage. The execution velocity is computed based on samples collected at periodic sampling intervals[1] by the System Resources Manager (SRM). The SRM sampling code interrogates address space control blocks (TCBs, SRBs, OUCBs, and OUXBs) to determine the state of each address space assigned to a service class. Sampling counts associated with the service class are updated based upon the state of the address spaces.

The sampling code records the sampling result into CPU using samples, CPU delay samples, CPU Capping delay samples, and Processor storage delay[2].

Notice that only certain delay categories are included: only delays for processor or for processor storage are included in the "delay" category. These delays are under control of the SRM. Delays not under control of the SRM are not included in CPU or processor storage delays, but are included in an "unknown" delay category. **Unknown delay is not included in the execution velocity computation**. The "unknown" delay means that the SRM was unable to identify the cause of delay. In practice, this means that the delay was something over which the SRM had no control (e.g., I/O operations, ENQ delay, etc.).

[1]With MVS/ESA SP5.1, the sampling interval is 250 milliseconds. The state of each TCB or SRB associated with an address space is sampled every 250 milliseconds, beginning from address space initiation.

[2]Processor storage delay samples means that an address space is ready to execute, but is delayed waiting for processor storage. Eight separate processor storage delays are recorded (swap-in delay, MPL delay, and six categories of page-in delay from auxiliary storage)

©Copyright 1996, Computer Management Sciences, Inc.

The Workload Manager computes the execution velocity of a service class by applying the following algorithm:

$$\frac{using\ samples}{using\ samples + delay\ samples} * 100$$

where:

using samples include:

- The number of samples of work using the processor (CPU Using).
- The number of calculated samples of work using non-paging DASD I/O resources (DASD connect state or DASD disconnect state). I/O using samples are included only if the installation has elected to include WLM-managed I/O.

delay samples include:

- The number of samples of work delayed for the processor (Denied CPU Delay or CPU Capping delay).
- The number of samples of work delayed for processor storage. Delay for processor storage includes:
    - Paging delay
    - Swap-in delay
    - Swapped out for multiprogramming (MPL) reasons
    - Server address space creation delay
    - Initiation delays for batch jobs in WLM-managed job classes
- The number of calculated samples of work delayed for non-paging DASD I/O resources (DASD IOS queue delay, DASD subchannel pending delay, or DASD control unit queue delay). I/O delay samples are included only if the installation has elected to include WLM-managed I/O.

The result from the algorithm is multiplied by 100, to yield an execution velocity ranging from 0 (when the address space did not use the CPU) to

100 (when the address space was not delayed for any reason controlled by the SRM).

It is important to keep in mind that execution velocity applies **only to times when an address space is using a CPU or ready to use a CPU**. It does not include times when an address space is idle, waiting for I/O, enqueued for a resource, etc. The SRM takes samples every 250 milliseconds, or 4 times per second. If the address spaces in a service class are idle or waiting for some unknown reason for most of the time, the SRM might not be able to collect sufficient samples to compute a valid execution velocity.

The following example illustrates the problem:

- Suppose that the address spaces in a service class are idle or waiting for unknown reasons for 95% of the time. This behavior is common with some Started Tasks (such as VTAM, RMF, etc.) During only 5% of the time, would the SRM find the address spaces in one of the states that contribute to execution velocity (Using CPU, CPU delay, processor storage delay).

- During the 10-second policy adjustment interval, the Workload Manager would have only 2 samples for the previous interval (4 samples per second * 10 seconds * 0.05 = 2).

- The Workload Manager normally keeps about 20 minutes history information. Over an entire 20 minute interval, the SRM would collect only 240 samples (20 minutes * 60 seconds per minute * 4 samples per second * 0.05 = 240).

- While 240 samples might be a sufficiently large number to yield a valid result, recall that this value is an accumulation over 20 minutes and Workload Manager decisions would necessarily assume that the 20 minutes' data represent the CPU demand and delays of the address spaces in the service class.

More insidious is the fact that, beginning with MVS/ESA SP3.1, the MVS Dispatcher algorithms were redesigned to implement a "reduced preemption" technique of dispatching[3]. With reduced preemption, a newly-ready task at a high dispatching priority might not immediately interrupt a task at a lower dispatching priority. Rather, dispatching operates on a "time-sliced" basis, and the interrupt might be delayed for a short time[4] before the Dispatcher proceeds with the interrupt. This algorithm was

[3]Lambourne (see reference) provides an excellent discussion of the full versus reduced/partial preemption algorithms.

[4]The delay is dynamically adjusted by the SRM, but typically varies between 1 and 5 milliseconds.

implemented to achieve greater benefit from the very high speed processor cache registers delivered with modern IBM processors.

- Reduced preemption has a significant effect on execution velocity. The tasks that are mostly idle (for example, Started Tasks) tend to use the processor is short bursts (that is, they are idle for a long percent of their elapsed time but want to use the processor when they become ready to execute). The tasks typically have a low mean time to wait when they are ready (that is, they use the CPU for a short time, then relinquish the CPU for I/O activity).

- If a Started Task uses only 100 microseconds of CPU time per dispatch and the average time between becoming ready and being dispatched is 2 milliseconds (because of reduced preemption), over 95% of the time the Started Task ready time would be waiting for CPU (CPU Delay). This time would translate into an **achievable** execution velocity of less than 5 (100÷(2000+100)=4.76), regardless of the execution velocity goal specified for the service class!

  The Workload Manager could not achieve a high execution velocity goal for this type of task, even though the Started Task had been assigned a high dispatch priority. This is an effect of the basic Dispatcher algorithms rather than the Workload Manager algorithms.

When CPExpert determines that a service class with an execution velocity goal has missed its performance goal, CPExpert reviews the number of samples upon which the execution velocity is based. CPExpert produces Rule WLM171 if the number of samples is small.

Once CPExpert has determined that an unacceptably small number of samples exist, no further analysis is done. It makes no sense to analyze delays to the service class based on a low number of samples, inasmuch as the conclusions from the samples would be invalid.

The following example illustrates the output from Rule WLM171:

```
RULE WLM171:  EXECUTION VELOCITY WAS COMPUTED ON A SMALL SAMPLE SET

   The delay information presented above is based on the CPU Using and
   Execution Delay samples of the ASCH Service Class (execution velocity
   is based on these samples).  These percentages show the distribution of
   time when an address space in the service class was executing (using the
   CPU, waiting to use the CPU, or waiting for processor storage).  For a
   significant percent of their overall active time, address spaces in the
   ASCH Service Class were either IDLE or were waiting on some event not
   included in the execution velocity calculations.  The below information
   shows the percent of total active time in which address spaces in this
   service class were executing, were delayed for UNKNOWN reasons, or were
   idle.  Please refer to Rule WLM171 in the WLM Component User Manual for
   discussion of the implications of this finding.

                          AVERAGE      PCT        PCT      PCT   EXECUTION
   MEASUREMENT INTERVAL     MPL     EXECUTING   UNKNOWN   IDLE    SAMPLES
   14:30-14:45,01MAR1994    1.0        0.0        0.0    100.0       3
   14:45-15:00,01MAR1994    1.0        0.1        0.0     99.9       5
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- If the service class reported by Rule WLM171 consists of Started Tasks, you should assess the important of the Started Tasks.

  - If the Started Tasks are important from a system view (e.g., VTAM), you should consider allowing the Started Tasks to default to the SYSSTC service class. The SYSSTC service class has a high dispatching priority. Address spaces in SYSSTC will not be subject to the Workload Manager's execution velocity algorithms[5].

  - If the Started Tasks are not important from a system view, you should consider removing them from a service class with relatively high execution velocity goals (since the Workload Manager is unable to achieve the goals). You may wish to assign them to a service class with (1) relatively low execution velocity goals or (2) discretionary goals.

- If you are comfortable with the current placement of address spaces in the service class reported by Rule WLM171, you should consider excluding the service class from analysis by CPExpert (using the EXCLUDE guidance parameters described in Section 2 (Chapter 1.1.8) of this document). You likely would become annoyed by CPExpert continually reporting that the service class missed its performance goal when you contemplate no action.

[5]This alternative does not reduce the effect of the reduced preemption on address spaces in the service class. The alternative simply removes them from the Workload Manager's control.

- Alternatively, you can provide different guidance to CPExpert's analysis by altering the EXECSAMP guidance variable in USOURCE(WLMGUIDE).

Reference: "MVS/ESA Full vs. Reduced/Partial Preemption", Lambourne, Steve, *1994 Proceedings of the Computer Measurement Group*, page 1347.

IBM TalkLink MVSWLM CFORUM, Appended at 19:23:56 on 10/24/96 by DISKER at KGNVMC (John Arwe, SRM/WLM Development Team).

©Copyright 1996, Computer Management Sciences, Inc.

Rule WLM172: Server was idle a significant percent of time

Finding: The service class period identified in Rule WLM104 or WLM105 missed its response goal. However, address spaces the server service class were Idle for a significant percent of their overall active time. Consequently, the Workload Manager delay information for the server service class may be meaningless.

Impact: This finding is provided for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101: Service Class did not achieve average response goal
Rule WLM102: Service Class did not achieve percentile response goal

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a transaction service class did not achieve its response performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. The delays from the served service class are reported by CICS/ESA Version 4.1 or IMS Version 5 interaction with the Workload Manager, using the Workload Management Services macros[1].

Please refer to Rule WLM120 to Rule WLM132 for a discussion of the delays from the served service class.

After analyzing the **served** service class delays, CPExpert identifies the **server** service class. The server service class normally will be one or more CICS regions or IMS regions. The subsystem service class (e.g., the CICS/ESA Version 4.1 region or IMS Version 5 region) must have a performance goal and importance defined, in order for the region to start-up. However, the performance goal and importance normally are used by the Workload Manager **only at start-up time** for the address space[2].

After start-up time, the Workload Manger normally ignores the goal and importance of subsystems. After start-up time, the Workload Manager

[1]Please refer to Section 4 of this document for more detail about the Workload Management Services macros and how the subsystems use these macros to exchange information with the Workload Manager.

[2]This statement is not true if the region should become idle for some period of time. If there are no transactions executing in the region for some time, the Workload Manager will rely on the performance goal and importance associated with the region to make resource allocation decisions. This situation should normally occur only during "off shifts" or for test regions with low activity.

©Copyright 1996, Computer Management Sciences, Inc.

normally uses the goal and importance of the "served" transaction service classes as the basis for it resource allocation decisions.

The Workload Manager attempts to meet the performance goals of the "served" transaction service classes. In order to meet these performance goals, the Workload Manager must assign resources to the server service class (e.g., the service class of the CICS region), regardless of the goal and importance assigned to the subsystem service class.

The Workload Manager periodically examines the SRM control blocks describing each address space and acquires samples[3] describing the state of each dispatchable unit of an address space (that is, each TCB or SRB associated with the address space). The Workload Manager accumulates the samples into counters which describe the state of the address space. The samples are summarized by service class period.

The analysis performed by the Workload Manager and subsequent analysis by CPExpert is based on samples. The reliability of sampling depends upon having a sufficiently large number of samples such that the samples represent the "population" being sampled[4]. If a small number of samples are taken, invalid conclusions might be reached based on an analysis of the samples. In order for the conclusions about causes of delays to be valid, sufficient samples must be taken while address spaces were in a "ready" state rather than in an "idle" state.

When CPExpert determines that a transaction service class has missed its performance goal, CPExpert reviews the number of samples taken during times when address spaces in the **server** were in a "ready" state. This number of samples is obtained by summing the CPU Using samples (R723CCUS), Total Delay samples (R723CTOT), and Unknown samples (R723CUNK). CPExpert produces Rule WLM172 if this total number of samples is small.

Once CPExpert has determined that an unacceptably small number of samples exist, no further analysis is done. It makes no sense to analyze delays to the service class based on a low number of samples, inasmuch as the conclusions from the samples would be invalid.

The following example illustrates the output from Rule WLM172:

[3]With MVS/ESA SP5.1 Goal Mode, the sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

[4]With the Workload Manager samples, the "population" consists of the possible execution states of address spaces being sampled.

```
RULE WLM172:  SERVER WAS IDLE A SIGNIFICANT PERCENT OF TIME

  The delay information presented above is based on the EXECUTION time of
  the CICSTEST server (the CPU Using, Execution Delay, and Unknown Delay).
  These percentages show the distribution of time while some transaction
  was active.  However, address spaces in the CICSTEST Service Class
  were IDLE for a significant percent of their overall active time.  The
  below information shows the percent of CICSTEST total active time in
  which address spaces were executing (processing transactions) or were
  idle.  Please refer to Rule WLM172 in the WLM Component User Manual for
  a discussion of the implications of this finding.

                          AVERAGE      PCT        PCT      PCT   EXECUTION
   MEASUREMENT INTERVAL     MPL     EXECUTING   UNKNOWN   IDLE    SAMPLES
   11:15-11:29,07DEC1994      1        0.1        .5      99.9        6
```

Suggestion: CPExpert suggests that you consider the following alternatives:

- You can ignore the finding (and previous rules in the logic flow) if you feel that the situation is unusual rather than a continuing status. For example, the finding might be made when a server was temporarily idle because development personnel were not submitting transactions to the CICS test region.

  If you chose to ignore the finding, you may wish to exclude the transaction service class from analysis, using the EXCLUDE guidance parameters described in Section 2 (Chapter 1.1.8) of this document). You likely would become annoyed by CPExpert continually reporting that the service class missed its performance goal when you contemplate no action.

- You may wish to delete the service class and assign the workload to a service class with more active address spaces if you feel that the situation is a continuing one. That is, if you feel that the address spaces normally are idle, you may wish to review whether they need their own service class. As general guidance, it is desirable to keep the service class periods to as small a number as possible.

Rule WLM173: The response performance goal may be too large

Finding: CPExpert believes that the response performance goal specified for a service class may be too large.

Impact: This finding should be viewed a LOW IMPACT or MEDIUM IMPACT on the performance of your computer system. The finding could have a HIGH impact on the performance of the service class identified by this finding.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Users specify a performance goal for each service class. There are four types of performance goals: average response, percentile response, execution velocity, and discretionary. The first two (average response and percentile response) are the subject of this rule description.

The Workload Manager ISPF Response Time Goal Panel allows a response performance goal of up to 24 hours to be specified. Response goals in minutes or hours are typically associated with batch workloads.

CPExpert believes that a response performance goal of over 5 minutes is likely to result in unsatisfactory performance in most environments and a response goal of **less than 1 minute** is more likely to yield desired results. The following discussion explains why CPExpert believes that relatively long response goals are inappropriate:

- The Workload Manager attempts to adjust system resources as necessary to achieve the performance goal specified for service classes. The Workload Manager evaluates how well the existing resource policy allows performance goals to be met **every 10 seconds**. This 10-second process is called the *Policy Adjustment Interval*.

  During policy adjustment, the Workload Manager evaluates the performance of each service class. The evaluation is accomplished by computing a Performance Index for each service class period and analyzing the Performance Index within each level of Goal Importance[1]

  Obviously, in order to analyze how well a service class is performing against a response goal, one or more "transactions" must have completed during the previous interval. If no transactions completed, the Workload Manager has no information on which to assess the

[1]Please refer to Section 4 for a more comprehensive discussion of the Policy Adjustment process.

©Copyright 1998, Computer Management Sciences, Inc.

performance of the service class with respect to the response goal. In fact, the process works much better if many transactions complete, as the Workload Manager can compute average response or percentile response based on a larger sample of work.

Once the Workload Manager makes a policy adjustment decision, it evaluates the effect of that decision during the **next** policy adjustment interval. In order to assess the effect of the decision on response time, multiple transactions must complete so the Workload Manager can evaluate the effect on transaction response[2] time.

This process works extremely well if the transactions represent interactive work (e.g., TSO transactions, CICS transactions, or IMS transactions). Many transactions normally will complete in a policy adjustment interval and the Workload Manager will have adequate information upon which to assess the results of the policy decisions.

If the "transaction" really is a batch job with a relatively long response performance goal, it is unlikely that many transactions will complete in the 10-second policy adjustment interval. Thus, the Workload Manager has little or no information on which to base its policy adjustment decisions; the Workload Manager must wait for batch jobs to complete before any decisions can be made. Thus, the Workload Manager will be unresponsive in adjusting system resources to meet the performance goal for the batch jobs.

- The Workload Manager evaluates system performance considering the performance of all service classes, based on their level of importance. Most modern computer environments have a mix of workload, consisting of both interactive and non-interactive. The interactive workload usually has a higher importance, and interactive workload often is quite dynamic in terms of system requirements.

  One consequence of this nature of interactive work is that the Workload Manager typically will adjust resource allocation policies based on the requirements of the interactive workload. Only in the most stable environments will policy adjustment decisions be drive by relatively lengthy non-interactive response goals.

- Workload Manager developers have stated that only 20 minutes of historical information are retained by the Workload Manager. At present,

[2]The Workload Manager can assess the effect of some policy decisions without response-related information. For example, suppose that the Workload Manager determined that paging was a major cause of performance degradation. The Workload Manager might make processor storage decisions to either protect or restrict central or expanded storage for certain service classes. The effect of these decisions would be apparent from a system view (e.g., paging increased or decreased) without requiring transaction response data. However, the Workload Manager cannot determine whether the overall response performance goal is being met until transactions complete.

it is unclear how the internal Workload Manager algorithms discard data and it is unclear what effect discarding data has on lengthy response goals.

- Please note that IBM's *Planning: Workload Management* specifically states "Work that is appropriate for a response goal should have a reasonable number of transaction completions over 20 minutes of time. If there are only a few completions, you are better off using a velocity goal."

There is an exception to this general advice. You may have defined service classes to describe subsystem transactions (such as CICS transactions) which have long Idle state times. Rule WLM122 describes transactions with long Idle state times, and suggests an approach which includes defining a **very long** response goal for the service class containing these transactions. CPExpert suppresses Rule WLM173 for transaction subsystem service classes.

The following example illustrates the output from Rule WLM173:

```
RULE WLM173:  THE RESPONSE PERFORMANCE GOAL MAY BE TOO LARGE

BATPRD (Period 1):  The service class had a response goal of 0:20:00:00.
This response goal is large relative to the intervals in which the
Workload Manager makes system adjustments.  The Workload Manager might
not have been  able to take effective actions with such a large goal
for the service class period.  You might have better success with an
execution velocity goal for this service class.  Please refer to Rule
WLM006 in the WLM Component User Manual for a discussion of this issue.

                             AVERAGE     AVERAGE ENDING TRANSACTIONS
MEASUREMENT INTERVAL        RESPONSE        PER 20 MINUTE INTERVAL
10:29-10:44,20JUL1998       31:04:22                     2
```

Suggestion: CPExpert suggests that you consider specifying an execution velocity goal for the service class identified by Rule WLM173.

You can adjust (or turn off) this analysis if you disagree with CPExpert's reasoning. The **MAXRESP** guidance variable in USOURCE(WLMGUIDE) can be used to provide guidance to CPExpert on the maximum response performance goal which CPExpert views as acceptable.

The default specification for the MAXRESP guidance variable in WLMGUIDE is %LET MAXRESP=0:05:00, indicating that CPExpert that any response performance goal greater than 5 minutes causes Rule WLM173 to be produced.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"Migrating to the MVS Workload Manager", Peter Enrico (IBM Corporation Workload Manager developer), 1995 SHARE Winter Meeting

©Copyright 1998, Computer Management Sciences, Inc.

Rule WLM200: Average CPU use per transaction is higher than goal

Finding: CPExpert has determined that the average CPU time per transaction was higher than the response goal for the service class. This finding does not apply to subsystem transactions (that is, it does not apply to CICS or IMS transactions).

Impact: This finding has a HIGH IMPACT on performance of your computer system.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101: Service Class did not achieve average response goal
Rule WLM102: Service Class did not achieve percentile response goal

Discussion: Transactions executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason. The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU. The resulting percentage is multiplied by the average transaction response time to yield the amount of time when the average transaction was using the CPU.

CPExpert compares the amount of time when the average transaction was using the CPU against the response goal. CPExpert produces Rule WLM200 if the CPU use per transaction is higher than the response goal.

©Copyright 1994, Computer Management Sciences, Inc.

The following example illustrates the output from Rule WLM200:

```
RULE WLM200: AVERAGE CPU USE PER TRANSACTION IS HIGHER THAN GOAL

   The average CPU time was higher than the response goal for Service
   Class ST_USERS (Period 1).  The average transaction used more CPU
   time than the response goal of 0.200.  MVS cannot achieve the
   response goal unless the CPU requirements of the average transaction
   can be reduced.  Alternatively, you can review the response goal to
   see whether the goal should be increased.  Please review the discussion
   with WLM200 regarding other alternatives.  This situation applies to the
   following measurement intervals:

                                     TOTAL           AVERAGE CPU TIME
   MEASUREMENT INTERVAL           TRANSACTIONS       PER  TRANSACTION
   14:00-14:15,01MAR1994               14                0.493
   14:15-14:30,01MAR1994               33                1.770
   14:30-14:45,01MAR1994               33                2.553
   14:45-15:00,01MAR1994              198                0.556
   15:00-15:16,01MAR1994               33                2.391
```

Suggestion: MVS cannot achieve the specified response goal for the service class unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM106 or Rule WLM107 (one of these rules will be produced depending upon the nature of the service class and performance goal).

  Determine whether most transactions missed the response objective or whether a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application code can be made more efficient, less CPU time will be required to process the transactions.

  If you find that some transactions skewed the findings, you may wish to consider other alternatives:

  - If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class.

You may wish to specify a different importance and different performance goal for this new service class.

- If you do not wish to place the transactions into a different service class (or are unable to identify them), perhaps you can establish another performance period for the existing service class. By specifying an appropriate DUR value, you can cause the SRM to migrate the transactions significantly using the CPU into a lower service class period (perhaps with a different importance and different performance goal).

  This particular alternative is easy to implement, and the inherent processing characteristics of the transactions will automatically cause them to be migrated to lower period service classes. As the CPU-intensive transactions use CPU cycles, they will accumulate service, and the SRM will migrate the CPU-intensive transactions to a lower performance period.

  This alternative is not listed as the initial alternative because the transactions will initially execute in Period 1 of the service class. By executing in Period 1 of the service class, the transactions may deprive short-running transactions of access to a processor and thus cause the short-running transactions to be unreasonably delayed.

- If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

  Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because they are using the CPU, not because they are denied access to the CPU[1].

[1]Although other rules also may show that transactions also are denied access to the CPU, Rule WLM200 reports that transactions are delayed because of CPU use.

Rule WLM201: Goal may be unrealistic - average CPU use per transaction is high

Finding: CPExpert has determined that the average CPU time per transaction was more than 75% of the response goal for the service class. This finding does not apply to subsystem transactions (that is, it does not apply to CICS or IMS transactions).

Impact: This finding has a HIGH IMPACT on the performance of your computer system.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101: Service Class did not achieve average response goal
Rule WLM102: Service Class did not achieve percentile response goal

Discussion: Transactions executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU.

- The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was using the CPU.

CPExpert compares the amount of time when the average transaction was using the CPU against the response goal. CPExpert produces Rule WLM200 if the CPU use per transaction is **higher** than the response goal. Otherwise, CPExpert produces Rule WLM201 if the CPU use per transaction is more than 75% of the response goal.

The following example illustrates the output from Rule WLM201:

```
RULE WLM201: GOAL MAY BE UNREALISTIC - AVERAGE CPU USE IS HIGH

   The average CPU time per transaction was high for Service Class
   TPNSODD (Period 6).  The CPU time of the average transaction
   was more than 75% of the response goal.  You may wish to review
   the application to see whether the CPU time can be reduced.
   Alternatively, you can review the response goal to see whether
   the goal should be increased.  Please review the discussion
   with WLM201 regarding other alternatives.  This situation applies
   to the following measurement intervals:

                                   TOTAL           AVERAGE CPU TIME
   MEASUREMENT INTERVAL         TRANSACTIONS       PER  TRANSACTION
   15:00-15:16,01MAR1994              58                0.809
```

Suggestion: MVS may not be able to achieve the specified response goal for the service class unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Determine whether this finding is appropriate for your installation and the transactions involved. The 75% was chosen arbitrarily as the default value, with the belief that you should be aware of such a significant amount of CPU use per transaction. You may find that the transactions naturally use a significant amount of CPU (rather than performing I/O or experiencing other delays). You can alter the "75%" default by using the HIGHCPU guidance variable in USOURCE(WLMGUIDE).

  However, keep in mind that this finding is produced only when a service class does not meet its response goal. If over 75% of the response time is attributed to CPU use, non-CPU related changes can address only the remaining 25% of response.

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application

code can be made more efficient, less CPU time will be required to process the transactions.

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM106 or Rule WLM107 (one of these rules will be produced depending upon the nature of the service class and performance goal). Determine whether most transactions missed the response objective or whether only a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

  If you find that a few transactions skewed the findings, you may wish to consider other alternatives:

  - If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

  - If you do not wish to place the transactions into a different service class (or are unable to identify them), perhaps you can establish another period for the existing service class. By specifying an appropriate DUR value, you can cause the SRM to migrate the transactions significantly using the CPU into a lower service class period (perhaps with a different importance and different performance goal).

    This particular alternative is easy to implement, and the inherent processing characteristics of the transactions will automatically cause them to be migrated to lower period service classes. As the CPU-intensive transactions use CPU cycles, they will accumulate service, and the SRM will migrate the CPU-intensive transactions to a lower performance period.

    This alternative is not listed as the initial alternative because the transactions will initially execute in Period 1 of the service class. By executing in Period 1 of the service class, the transactions may deprive short-running transactions of access to a processor and thus cause the short-running transactions to be unreasonably delayed.

  - If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather,

the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

  Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because they are using the CPU, not because they are denied access to the CPU[1].

[1]Although other rules also may show that transactions also are denied access to the CPU, Rule WLM201 reports that transactions are delayed because of CPU use.

Rule WLM202: Average CPU use was a major cause of transaction delay

Finding: CPExpert has determined that the average CPU time per transaction was a major cause of transaction delay. This finding does not apply to subsystem transactions (that is, it does not apply to CICS or IMS transactions).

Impact: The impact of this finding depends upon the amount of CPU use by the service class. A high percent of CPU use means HIGH IMPACT while low percent of CPU use means LOW IMPACT. See the output associated with the rule which caused this rule to be invoked (Rule WLM101 or Rule WLM102, depending upon the type of service class and performance goal).

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal

Discussion: Transactions executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason. The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU.

©Copyright 1994, Computer Management Sciences, Inc.

- The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was using the CPU.

CPExpert compares the amount of time when the average transaction was using the CPU against the response goal. CPExpert produces Rule WLM200 if the CPU use per transaction is **higher** than the response goal. Otherwise, CPExpert produces Rule WLM201 if the CPU use per transaction is more than 75% of the response goal.

If neither Rule WLM200 nor Rule WLM201 are produced, CPExpert determines whether CPU use was a **primary** or **secondary** cause of a service class not meeting its response goal. CPExpert produces Rule WLM202 if CPU use was a primary or secondary cause.

The following example illustrates the output from Rule WLM202:

```
RULE WLM202: AVERAGE CPU USE WAS A MAJOR CAUSE OF TRANSACTION DELAY

   The CPU time used by the application was a major delay to the average
   transaction in Service Class TPNSODD (Period 6).  You may wish to
   review the application to see whether the CPU time can be reduced.
   Alternatively, you can review the response goal to see whether the
   goal should be increased.  Please review the discussion with WLM202
   regarding other alternatives.  This situation applies to the following
   measurement intervals:

                                   TOTAL           AVERAGE CPU TIME
   MEASUREMENT INTERVAL         TRANSACTIONS       PER  TRANSACTION
   15:00-15:16,01MAR1994              58                0.809
```

Suggestion: CPExpert has determined that CPU use is the primary or secondary cause of the service class not achieving its response goal. MVS may not be able to achieve the performance goal unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application code can be made more efficient, less CPU time will be required to process the transactions.

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM106 or Rule

WLM107 (one of these rules will be produced depending upon the nature of the service class and performance goal). Determine whether most transactions missed the response objective or whether only a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

If you find that a few transactions skewed the findings, you may wish to consider other alternatives:

- If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

- If you do not wish to place the transactions into a different service class (or are unable to identify them), perhaps you can establish another period for the existing service class. By specifying an appropriate DUR value, you can cause the SRM to migrate the transactions significantly using the CPU into a lower service class period (perhaps with a different importance and different performance goal).

  This particular alternative is easy to implement, and the inherent processing characteristics of the transactions will automatically cause them to be migrated to lower period service classes. As the CPU-intensive transactions use CPU cycles, they will accumulate service, and the SRM will migrate the CPU-intensive transactions to a lower performance period.

  This alternative is not listed as the initial alternative because the transactions will initially execute in Period 1 of the service class. By executing in Period 1 of the service class, the transactions may deprive short-running transactions of access to a processor and thus cause the short-running transactions to be unreasonably delayed.

- If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- If the service class has multiple periods, and if this service class period is not the <u>last</u> period, you may wish to consider revising the

duration of the service class period in which the transaction is executing. Perhaps by specifying a smaller duration value, users of a relatively large amount of CPU service would be moved to a lower service class period more quickly. The remaining transactions in the service class period might then meet the response goal specified.

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

  Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because they are using the CPU, not because they are denied access to the CPU[1].

[1]Although other rules also may show that transactions also are denied access to the CPU, Rule WLM201 reports that transactions are delayed because of CPU use.

Rule WLM210: Average Server CPU use per transaction is higher than goal

Finding: CPExpert has determined that the average server CPU time per transaction was higher than the response goal for the service class.

Impact: This finding has a HIGH IMPACT on performance of your computer system.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120(series) and Rule WLM130(series) describe the results from this analysis.

After analyzing the subsystem transaction delays, CPExpert identifies the service classes which serve the transactions. The subsystem transactions typically are CICS transactions, and the servers are the CICS regions. Alternatively, the transactions could be IMS transactions and the servers could be the IMS control regions or transaction processing regions.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS. CPExpert analyzes the CPU requirements of the server service class to determine whether the transaction required a significant amount of CPU time.

CPU usage and other resource requirements are not contained in the SMF Type 72 records which describe the subsystem transaction service class.

©Copyright 1994, Computer Management Sciences, Inc.

These subsystem transaction service classes are not address spaces, but are logical groupings of transactions. Resource information is **not** recorded by SMF for the transactions, but the resource information **is** recorded for the address spaces (the servers) providing service to the service classes. Consequently, CPExpert analyzes the resource requirements of the server service classes.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU. The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was using the CPU.

- Server service classes might serve multiple subsystem service classes. For example, a CICS region (the server) might provide service to a number of service classes which describe different CICS transactions. If the server provides service to multiple service classes, the above technique automatically "pro-rates" the CPU requirements of the server. This automatic "pro-rating" is possible because the sampling process of the SRM is independent of the transactions executing.

CPExpert produces Rule WLM210 if the average CPU time per transaction is higher than the performance goal for the transaction service class. With Rule WLM210, CPExpert shows the percent of total service provided by the server to the transaction service class missing its performance goal. This value is computed by dividing the number of times an address space in the server provided service to the transaction service class, by the total number of times the server provided service to all transaction service classes.

If CPExpert is analyzing served transactions from the Execution Phase view, and if no transactions ended Execution Phase, CPExpert will produce “???” in the TRANS column. Otherwise, CPExpert will print the number of transactions that completed Execution Phase.

If CPExpert is analyzing served transactions from the Begin_to_end (BTE) Phase view, and if no transactions ended in the BTE Phase, CPExpert will produce “???” in the TRANS column. Otherwise, CPExpert will print the number of transactions that completed the BTE Phase.

The following example illustrates the output from Rule WLM210:

```
RULE WLM210: AVERAGE SERVER CPU USE PER TRANSACTION IS HIGHER THAN GOAL

  The average CPU time per transaction by the server (CICSRGN) was higher
  than the response goal for Service Class CICUSRTX.  If CICSRGN provided
  service to more than one service class, CPExpert prorated the CPU time
  based on the number of times that CICSRGN provided service to CICUSRTX.
  Using these calculations, the average transaction used more CPU time
  than the response goal of CICUSRTX.  This situation applies to the
  following RMF measurement intervals:

                                TOTAL      AVG SERVER CPU         PCT
  MEASUREMENT INTERVAL          TRANS      TIME PER TRANS       SERVICE
  13:07-13:12,21JUN1994        14,307       0:00:00.836          99.5
  13:17-13:22,21JUN1994        14,314       0:00:00.834          99.7
```

Suggestion; MVS cannot achieve the specified response goal for the service class unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM108 or Rule WLM109 (one of these rules will be produced depending upon the nature of the service class and performance goal). Determine whether most transactions missed the response objective or whether a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

  If you find that some transactions skewed the findings, you may wish to consider other alternatives:

  - If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

  - If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather,

the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application code can be made more efficient, less CPU time will be required to process the transactions.

- Review the CPU requirements of the server (either the CICS region or the IMS region).

  If the server is CICS, you should execute the CPExpert CICS Component against the CICS region to identify performance improvement opportunities. If you have not licensed the CPExpert CICS Component, you should follow the "Processor Cycles Checklist" in IBM's *CICS Performance Guide*.

  If the server is an IMS region, IBM suggests the following actions to reduce CPU time used by the IMS region[1]:

  > "The total number of machine instructions that are executed to process a transaction, including system services, IMS services, and the application program itself, has a direct bearing on throughput. The accumulation of executed instructions is termed the path length. The actions suggested in the previous IMS Options section all contribute to the minimization of path length.
  > "Avoid the regular use of traces such as the DL/I Call Image Capture and other traces invoked by the /TRACE command. These are specified as parameters on the OPTIONS statement in the DFSVSMxx member of IMS.PROCLIB.
  >
  > "Do not run the IMS Monitor (DFSMNTR0), except for 10- to 20-minute preplanned intervals.
  >
  > "In a real-storage constrained system, the most effective way to reduce path length is to minimize paging. Minimal pools contribute to minimize paging by eliminating costly scanning of directories or buffers which might have to be paged in before they can be read. If virtual storage requirements are reduced:
  >
  > - A minimal PSB pool minimizes buffer searching.

[1]Source: *IMS/ESA Version 4: System Administration Guide*, Section 7.2.6 Minimizing Path Length (BookManager document).

- A tuned database pool minimizes buffer searching; a larger database pool costs more in path length and might not reduce I/O.

- A tuned message queue pool minimizes buffer searching; a larger pool reduces IMS message queue I/O but at the expense of a higher processor cycles per queue pool operation.

- The same applies to the message format pool as to the message queue pool."

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

  Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because the server service class is using the CPU, not because the server is denied access to the CPU[2].

[2]Although other rules may show that transactions also are denied access to the CPU, Rule WLM210 reports that subsystem transactions are delayed because of the amount of CPU use by the server service class.

Rule WLM211: Goal may be unrealistic - average Server CPU use per transaction is high

Finding: CPExpert has determined that the average server CPU time per transaction was more than 75% of the response goal for the transaction service class.

Impact: This finding has a HIGH IMPACT on performance of your computer system.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120(series) and Rule WLM130(series) describe the results from this analysis.

After analyzing the subsystem transaction delays, CPExpert identifies the service classes which serve the transactions. The subsystem transactions typically are CICS transactions, and the servers are the CICS regions. Alternatively, the transactions could be IMS transactions and the servers could be the IMS control regions or transaction processing regions.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS. CPExpert analyzes the CPU requirements of the server service class to determine whether the transaction required a significant amount of CPU time.

©Copyright 1994, Computer Management Sciences, Inc. Revised: October, 2001

CPU usage and other resource requirements are not contained in the SMF Type 72 records which describe the subsystem transaction service class. These subsystem transaction service classes are not address spaces, but are logical groupings of transactions. Resource information is not recorded by SMF for the transactions, but is recorded for the address spaces (the servers) providing service to the service classes. Consequently, CPExpert analyzes the resource requirements of the server service classes.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU. The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was using the CPU.

- Server service classes might serve multiple subsystem service classes. For example, a CICS region (the server) might provide service to a number of service classes which describe different CICS transactions. If the server provides service to multiple service classes, the above technique automatically "pro-rates" the CPU requirements of the server. This automatic "pro-rating" is possible because the sampling process of the SRM is independent of the transactions executing.

CPExpert compares the amount of time when the average transaction was using the CPU against the response goal. CPExpert produces Rule WLM210 if the CPU use per transaction is **higher** than the response goal. Otherwise, CPExpert produces Rule WLM211 if the CPU use per transaction is more than 75% of the response goal.

Suggestion: MVS may not be able to achieve the specified response goal for the service class unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Determine whether this finding is appropriate for your installation and the transactions involved. The 75% was chosen arbitrarily as the default value, with the belief that you should be aware of such a significant amount of CPU use per transaction. You may find that the transactions naturally use a significant amount of CPU (rather than performing I/O or experiencing other delays). You can alter the "75%" default by using the HIGHCPU guidance variable in USOURCE(WLMGUIDE).

  However, keep in mind that this finding is produced only when a service class does not meet its response goal. If over 75% of the response time is attributed to CPU use, non-CPU related changes can address only the remaining 25% of response.

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application code can be made more efficient, less CPU time will be required to process the transactions.

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM108 or Rule WLM109 (one of these rules will be produced depending upon the nature of the service class and performance goal). Determine whether most transactions missed the response objective or whether only a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

  If you find that a few transactions skewed the findings, you may wish to consider other alternatives:

  - If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

  - If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- Review the CPU requirements of the server (either the CICS region or the IMS region).

  If the server is CICS, you should execute the CPExpert CICS Component against the CICS region to identify performance improvement opportunities. If you have not licensed the CPExpert CICS Component, you should follow the "Processor Cycles Checklist" in IBM's *CICS Performance Guide*.

  If the server is an IMS region, IBM suggests the following actions to reduce CPU time used by the IMS region[1]:

  > "The total number of machine instructions that are executed to process a transaction, including system services, IMS services, and the application program itself, has a direct bearing on throughput. The accumulation of executed instructions is termed the path length. The actions suggested in the previous IMS Options section all contribute to the minimization of path length.
>
> "Avoid the regular use of traces such as the DL/I Call Image Capture and other traces invoked by the /TRACE command. These are specified as parameters on the OPTIONS statement in the DFSVSMxx member of IMS.PROCLIB.
>
> "Do not run the IMS Monitor (DFSMNTR0), except for 10- to 20-minute preplanned intervals.
>
> "In a real-storage constrained system, the most effective way to reduce path length is to minimize paging. Minimal pools contribute to minimize paging by eliminating costly scanning of directories or buffers which might have to be paged in before they can be read. If virtual storage requirements are reduced:
>
> - A minimal PSB pool minimizes buffer searching.
> - A tuned database pool minimizes buffer searching; a larger database pool costs more in path length and might not reduce I/O.
> - A tuned message queue pool minimizes buffer searching; a larger pool reduces IMS message queue I/O but at the expense of a higher processor cycles per queue pool operation.

[1]Source: *IMS/ESA Version 4: System Administration Guide*, Section 7.2.6 Minimizing Path Length (BookManager document).

- The same applies to the message format pool as to the message queue pool."

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

  Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because the server service class is using the CPU, not because the server is denied access to the CPU[2].

[2]Although other rules may show that transactions also are denied access to the CPU, Rule WLM210 reports that subsystem transactions are delayed because of the amount of CPU use by the server service class.

Rule WLM212: Average CPU use was a major cause of transaction delay

Finding: CPExpert has determined that the average CPU time per transaction was a major cause of transaction delay.

Impact: The impact of this finding depends upon the amount of CPU use by the service class. A high percent of CPU use means HIGH IMPACT while low percent of CPU use means LOW IMPACT. See the output associated with Rule WLM104 or Rule WLM105, depending upon the type of service class and performance goal.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120 and Rule WLM132 describe the results from this analysis.

After analyzing the subsystem transaction delays, CPExpert identifies the service classes which serve the transactions. The subsystem transactions typically are CICS transactions, and the servers are the CICS regions. Alternatively, the transactions could be IMS transactions and the servers could be the IMS control regions or transaction processing regions.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS. CPExpert analyzes the CPU requirements of the server service class to determine whether the transaction required a significant amount of CPU time.

©Copyright 1994, Computer Management Sciences, Inc.

CPU usage and other resource requirements are not contained in the SMF Type 72 records which describe the subsystem transaction service class. These subsystem transaction service classes are not address spaces, but are logical groupings of transactions. Resource information is not recorded by SMF for the transactions, but is recorded for the address spaces (the servers) providing service to the service classes. Consequently, CPExpert analyzes the resource requirements of the server service classes.

CPExpert analyzes the amount of CPU time used by transactions by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Using samples (R723CCUS) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was using the CPU. The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was using the CPU.

- Server service classes might serve multiple subsystem service classes. For example, a CICS region (the server) might provide service to a number of service classes which describe different CICS transactions. If the server provides service to multiple service classes, the above technique automatically "pro-rates" the CPU requirements of the server. This automatic "pro-rating" is possible because the sampling process of the SRM is independent of the transactions executing.

CPExpert compares the amount of time when the average transaction was using the CPU against the response goal. CPExpert produces Rule WLM210 if the server CPU use per transaction is **higher** than the response goal. Otherwise, CPExpert produces Rule WLM211 if the server CPU use per transaction is more than 75% of the response goal.

If neither Rule WLM210 nor Rule WLM211 are produced, CPExpert determines whether the server CPU use was a significant cause of the transaction service class not meeting its performance goal. CPExpert produces Rule WLM212 if the server CPU use was a significant cause of the transaction service class not meeting its performance goal.

Suggestion: CPExpert has determined that server CPU use was a significant cause of the transaction service class not meeting its response goal. MVS may not be able to achieve the performance goal unless the CPU requirements of the average transaction can be reduced.

CPExpert suggests that you consider the following actions:

- Review your performance goal for the transactions served by the service class, to determine whether the response goal is correct.

- Review the application processing the transactions, to determine whether the application code can more efficiently use the CPU. If the application code can be made more efficient, less CPU time will be required to process the transactions.

- Perform a "reality" check on the finding from CPExpert by examining the "Response Time Distribution" produced by Rule WLM108 or Rule WLM109 (one of these rules will be produced depending upon the nature of the service class and performance goal). Determine whether most transactions missed the response objective or whether only a few transactions **significantly** missed the response objective. If only a few transactions **significantly** missed the response objective, it is likely that these transactions skewed the findings.

  If you find that a few transactions skewed the findings, you may wish to consider other alternatives:

  - If you can identify the transactions, perhaps you can use Workload Categorization to place the transactions into a different service class. You may wish to specify a different importance and different performance goal for this new service class.

  - If you have specified an **average response goal** for the service class, perhaps you can change the goal to a **percentile response goal**. With a percentile goal, the Workload Manager would not be as concerned about the few transactions which used significantly more resources and consequently skewed the average response. Rather, the Workload Manager would base its workload management decisions on the percent of transactions which met the response goal.

- Review the CPU requirements of the server (either the CICS region or the IMS region).

  If the server is CICS, you should execute the CPExpert CICS Component against the CICS region to identify performance improvement opportunities. If you have not licensed the CPExpert CICS Component,

you should follow the "Processor Cycles Checklist" in IBM's *CICS Performance Guide*.

If the server is an IMS region, IBM suggests the following actions to reduce CPU time used by the IMS region[1]:

> "The total number of machine instructions that are executed to process a transaction, including system services, IMS services, and the application program itself, has a direct bearing on throughput. The accumulation of executed instructions is termed the path length. The actions suggested in the previous IMS Options section all contribute to the minimization of path length.
>
> "Avoid the regular use of traces such as the DL/I Call Image Capture and other traces invoked by the /TRACE command. These are specified as parameters on the OPTIONS statement in the DFSVSMxx member of IMS.PROCLIB.
>
> "Do not run the IMS Monitor (DFSMNTR0), except for 10- to 20-minute preplanned intervals.
>
> "In a real-storage constrained system, the most effective way to reduce path length is to minimize paging. Minimal pools contribute to minimize paging by eliminating costly scanning of directories or buffers which might have to be paged in before they can be read. If virtual storage requirements are reduced:
>
> - A minimal PSB pool minimizes buffer searching.
> - A tuned database pool minimizes buffer searching; a larger database pool costs more in path length and might not reduce I/O.
> - A tuned message queue pool minimizes buffer searching; a larger pool reduces IMS message queue I/O but at the expense of a higher processor cycles per queue pool operation.
> - The same applies to the message format pool as to the message queue pool."

- If none of the above options are applicable, and if this service class is very important, you may wish to consider running the application on a more powerful processor.

[1]Source: *IMS/ESA Version 4: System Administration Guide*, Section 7.2.6 Minimizing Path Length (BookManager document).

Note that simply increasing the Importance specified to the Workload Manager, or adding more logical processors (in an LPAR environment) will not resolve the problem with the service class not achieving its response goal. Transactions are delayed because the server service class is using the CPU, not because the server is denied access to the CPU[2].

[2]Although other rules may show that transactions also are denied access to the CPU, Rule WLM210 reports that subsystem transactions are delayed because of the amount of CPU use by the server service class.

Rule WLM220: Service Class was delayed because of resource capping

Finding: CPExpert has determined that resource capping was a major cause of the service class not achieving its performance goal.

Impact: The impact of this finding depends upon the amount of resource capping delay experienced by the service class. A high percent of resource capping delay means HIGH IMPACT while a low percent of resource capping means LOW IMPACT. See the output associated with the rule which caused this rule to be invoked (Rule WLM101 to Rule WLM103, depending upon the type of service class and performance goal).

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal

Discussion: Resource capping is a way of controlling the distribution of CPU service to one or more service classes. Resource capping is implemented by defining "resource groups" to the Workload Manager. A resource group is simply a named set of two values: a minimum CPU service specification and a maximum CPU service specification. The specifications are in terms of **unweighted CPU service units** (that is, the CPU service coefficients are not applied to TCB nor SRB raw CPU service units).

The Workload Manager will attempt to provide the minimum CPU service to the resource group and will restrict the resource from using more than the maximum CPU service.

Service classes are associated with resource groups; however, a particular service class can be associated with only one resource group[1].

It normally is not advisable to use resource groups. IBM provides the facility solely for special cases, and IBM does not contemplate resource groups being normally used. Resource group specifications are "preemptive" in nature, in that the Workload Manager attempts to honor resource group specifications before considering other service

[1]Please see Section 4 (Chapter 1.6) for a discussion of resource groups and how the Workload Manager implements the resource group specifications.

specifications. Consequently, **resource group specifications could nullify the rest of the Workload Manager's algorithms**.

When the maximum CPU service specified in the resource group has been used, the Workload Manager marks "non-dispatchable" the TCBs and SRBs associated with the service classes assigned to the resource group. This is the situation addressed by Rule WLM220.

As the System Resources Manager takes its samples of the state of address spaces, it examines whether a dispatchable unit (TCB or SRB) is marked non-dispatchable because of a resource group maximum. Samples reflecting the resource group maximum are recorded by RMF in the SMF Type 72 delay samples, as CPU Capping Delay (R723CCCA).

CPExpert computes the percent of CPU Capping Delay for the service class, as a function of the overall execution of transactions executing in the service class. CPExpert produces Rule WLM220 if the percent of CPU Capping Delay for the service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

With Rule WLM220, CPExpert provides the total number of ending transactions in the RMF measurement interval, the total CPU service units consumed by the transactions, the average CPU service units per transaction, and the average percent resource capping delay to transactions active in the service class.

The following example illustrates the output from Rule WLM220:

```
RULE WLM220: SERVICE CLASS WAS DELAYED BECAUSE OF RESOURCE CAPPING

   Service Class BATCH (Period 1) was delayed waiting for CPU
   resource capping.  This means that a TCB or SRB in the Service Class
   was marked non-dispatchable because the Resource Group maximum was
   being enforced.  Service Class BATCH (Period 1) was assigned
   Resource Group BATCHCAP, which specified a maximum of 500 CPU
   service units per second.  This situation applies to the following
   measurement intervals:

                                                                  AVG %
                            TOTAL     TOTAL CPU      AVERAGE CPU  CAPPING
   MEASUREMENT INTERVAL     TRANS   SERVICE UNITS   SERVICE UNITS  DELAY
   15:00-15:16,01MAR1994      658      452,664           876        22.1
```

Suggestion: As mentioned above, resource groups are intended for very special situations. In most environments, it is far better to allow the Workload Manager to manage system resources to meet the performance goals specified for various service classes. Using resource groups takes control away from the Workload Manager.

Further, specifying maximum CPU service units may result in unused CPU capacity if there are no other service classes ready to use the CPU service.

CPExpert suggests that you consider the following alternatives:

- While there may be unusual situations in which control must be removed from the Workload Manager, please consider whether you have such an unusual situation. If you do not have an unusual situation, you may wish to remove the resource group specification from the service class definition.

- Alternatively, you should review the performance goal specified for the service class identified by Rule WLM220. CPExpert performs "delay analysis" only on service classes which fail to achieve their performance goal. Consequently, the service class identified by Rule WLM220 had failed to achieve its performance goal.

  The performance goal may be incompatible with the resource group Capacity Maximum, and you may wish to either increase the performance goal (for response goals) or decrease the performance goal (for execution velocity goals).

- Alternatively, you should review the CPU usage report produced by CPExpert at the end of the normal rule listing. Compare the CPU time used by the service class identified by Rule WLM220 with the CPU time used by other service classes. Pay particular attention to CPU time used by any service classes at the same or lower importance, to see whether these service classes should receive the CPU service indicated.

- Alternatively, you may wish to increase the Capacity Maximum specified for the resource group. Since applications executing in the service class are being delayed because of CPU capping, you may remove or decrease the delay by increasing the Capacity Maximum for the resource group.

- Alternatively, you may wish to review the applications executing in the service class identified by Rule WLM220, to determine whether the application code can be optimized so that less CPU time is required.

- If none of the above alternatives apply and if Rule WLM220 continually is produced for the service class, you may wish to exclude the service class from CPExpert's analysis[2]. There is little point in having findings produced which cannot be acted upon.

[2]Please see Section 2 for information on how to exclude service classes from analysis.

Rule WLM221: Service Class was capped for discretionary goal management

Finding: CPExpert has determined that resource capping was a major cause of the service class not achieving its performance goal, but the service class had been capped for discretionary goal management.

Impact: The impact of this finding depends upon the amount of resource capping delay experienced by the service class.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal

Discussion: Resource capping is a way of controlling the distribution of CPU service to one or more service classes. Resource capping normally is implemented by defining "resource groups" to the Workload Manager. A resource group is simply a named set of two values: a minimum CPU service specification and a maximum CPU service specification. The specifications are in terms of **unweighted CPU service units** (that is, the CPU service coefficients are not applied to TCB nor SRB raw CPU service units).

The Workload Manager will attempt to provide the minimum CPU service to the resource group and will restrict the resource from using more than the maximum CPU service.

When the maximum CPU service specified in the resource group has been used, the Workload Manager marks "non-dispatchable" the TCBs and SRBs associated with the service classes assigned to the resource group.

This performance issue caused by normal resource capping is addressed by Rule WLM220. Rule WLM221 (this rule) addresses a slightly different issue.

A problem existed when using discretionary goals prior to OS/390 Version 2 Release 6: on systems in which 100% of the CPU was used by service class periods with performance goals, service class periods assigned a discretionary goal might never receive CPU service. This situation existed even though the service class periods with performance goals might be significantly **overachieving** their goals, since the Workload Manager would

©Copyright 1998, Computer Management Sciences, Inc.

never allow discretionary work to have a CPU dispatching priority equal to or higher than work with performance goals.

From one perspective, this algorithm is proper; discretionary work is defined as work that has no performance goal. However, most sites want the discretionary work eventually to be processed, even though it has no performance goal. Consequently, many sites removed the discretionary goal from work and assigned a performance goal to the work.

However, there are significant advantages to assigning a discretionary goal to work: work with a discretionary goal executes with the Mean-Time-To-Wait (MTTW) algorithm.

- Work assigned to a Mean-Time-To-Wait group competes within the Mean-Time-To-Wait group for access to the processor. Address spaces are assigned dispatching priority within the MTTW group, based upon their execution characteristics. Address spaces that execute a significant amount of CPU instructions between I/O operations are considered heavy CPU users. These heavy users receive a lower dispatching priority within the MTTW group than do address spaces requiring less CPU processing between I/O operations.

- The philosophy behind assigning work to Mean-Time-To-Wait groups is to attempt to use as much of the overall computer system as possible. Dispatching relatively light CPU users ahead of relatively heavy CPU users ensures that the I/O complex will be used simultaneously with the CPU processor. Since both CPU and I/O are active simultaneously, more overall work will be accomplished by the computer system. This philosophy assumes, of course, that overall throughput is a major goal, rather than the turnaround of specific heavy CPU users. This philosophy is explicitly applicable to service class periods assigned a discretionary goal.

IBM addressed this problem in OS/390 Version 2 Release 6, by implementing the *discretionary goal management* algorithms .

With discretionary goal management, the Workload Manager identifies service class periods that have been assigned a performance goal and which are candidates for participation in discretionary goal management. Service class periods can participate in discretionary goal management if either of the following conditions apply:

- The service class period has a response goal greater than one minute. This condition does not apply to subsystem transaction service classes (e.g., CICS or IMS transaction service classes), since these service class periods do not include address spaces.

- The service class period has an execution velocity goal less than or equal to 30%.

The Workload Manager identifies candidate service class periods meeting either of the above conditions, that have **significantly** overachieved their performance goal. If discretionary work exists in the system, the Workload Manager may apply *internal resource capping* to the service class periods that are overachieving their performance goal. The internal resource capping operates similarly to the normal Resource Group capping described in Chapter 1.6 of this section, in that the Workload Manager will cap the address spaces for one or more cap slices. This capping restricts the amount of CPU service that can be used by address spaces in the capped service class period.

The Workload Manager may apply internal resource capping when the Performance Index is less than 0.7, and stops internal resource capping when the Performance Index is greater than or equal to 0.81. If a candidate service class period with a performance goal has multiple periods, later periods are selected for capping before earlier periods (that is, capping would potentially be applied to Period 2 before capping would be considered for Period 1).

The effect of the discretionary goal management algorithm is to allow discretionary work to receive CPU cycles when work with a performance goal would otherwise significantly overachieve its performance goal.

As the System Resources Manager takes its samples of the state of address spaces, it examines whether a dispatchable unit (TCB or SRB) is marked non-dispatchable because of a resource group maximum. Samples reflecting the resource group maximum are recorded by RMF in the SMF Type 72 delay samples, as CPU Capping Delay (R723CCCA).

CPExpert computes the percent of CPU Capping Delay for the service class, as a function of the overall execution of transactions executing in the service class.

- CPExpert produces Rule WLM220 if the percent of CPU Capping Delay for the service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE) **and** the service class had been assigned to a Resource Group.

- CPExpert produces Rule WLM221 if the percent of CPU Capping Delay for the service class is greater than the significance value specified in the **WLMSIG** guidance variable in

USOURCE(WLMGUIDE) and the service class had **NOT** been assigned to a Resource Group.

With Rule WLM221, CPExpert provides the total number of ending transactions in the RMF measurement interval, the total CPU service units consumed by the transactions, the average CPU service units per transaction, and the average percent resource capping delay to transactions active in the service class.

The following example illustrates the output from Rule WLM221:

```
RULE WLM221:SERVICE CLASS WAS CAPPED FOR DISCRETIONARY GOAL MANAGEMENT

   Service Class BATCHLO (Period 2) was delayed waiting for CPU
   resource capping.  This means that a TCB or SRB in the Service Class
   was marked non-dispatchable because the Resource Group maximum was
   being enforced. The service class was not assigned to a Resource
   Group, but the Workload Manager implemented internal resource
   capping as a part of discretionary goal management.  Normally,
   this will not be a concern (as the WLM will not implement internal
   resource capping unless the service class period is over-achieving
   its goal).

                                                                  AVG %
                            TOTAL     TOTAL CPU      AVERAGE CPU  CAPPING
   MEASUREMENT INTERVAL     TRANS   SERVICE UNITS   SERVICE UNITS  DELAY
   15:00-15:16,01MAR1994      8        36,892           4611        14.4
```

Suggestion: This finding normally should not be produced, as explained in the above discussion; the Workload Manager will not select a service class period with a performance goal for internal resource group capping unless the service class period is significantly overachieving its performance goal.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals

"Pop the Hood on Workload Manager", Steve Grabarits and Gail Whistance (IBM Corporation Workload Manager developers), Session 2513, SHARE Technical Conference, August 1998.

Rule WLM222: Service Class was Active, but server was CPU capped

Finding: CPExpert has determined that resource capping was a major cause of the service class not achieving its performance goal.

Impact: The impact of this finding depends upon the amount of resource capping delay experienced by the service class. A high percent of resource capping delay means HIGH IMPACT while a low percent of resource capping means LOW IMPACT. See the output associated with the rule which caused this rule to be invoked (Rule WLM104 or Rule WLM105, depending upon the type of service class and performance goal).

Logic flow: The following rules cause this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state
Rule WLM121: Significant transaction time was in Ready state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120(series) to Rule WLM130(series) describe the results from this analysis.

After analyzing the subsystem transaction delays, CPExpert identifies the service classes which serve the transactions. The subsystem transactions typically are CICS transactions, and the servers are the CICS regions. Alternatively, the transactions could be IMS transactions and the servers could be the IMS control regions or transaction processing regions.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records. Please see Section 4 for a discussion of these states and the sampling process.

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The

©Copyright 1994, Computer Management Sciences, Inc.

Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS.

CPExpert produces Rule WLM121 when a significant cause of delay to a subsystem transaction was that the transaction was in Ready state. The ready state indicates that there was a program ready to execute on behalf of a work request in the "served" service class, but that the work manager has given priority to another work request. In the case of a CICS region, this means that there were more CICS tasks ready to execute in the "served" service class than were dispatched by CICS.

When Rule WLM120 or WLM121 are produced, CPExpert analyzes the CPU requirements and CPU capping of the server service class to determine whether the transaction required a significant amount of CPU time (which might be indicated by the Active state) or whether transactions were delayed in the Ready state (because the CICS region had been CPU capped).

CPU usage and other resource requirements are not contained in the SMF Type 72 records which describe the subsystem transaction service class. These subsystem transaction service classes are not address spaces, but are logical groupings of transactions. Resource information is not recorded by SMF for the transactions, but is recorded for the address spaces (the servers) providing service to the service classes. Consequently, CPExpert analyzes the resource requirements of the server service classes.

CPExpert analyzes the amount of CPU capping time of the server service classes by the following process:

- CPExpert first computes the number of samples which found an address space executing in the service class. This is done by summing CPU Using samples (R723CCUS), Total Wait samples (R723CTOT), and Unknown Delay samples (R723CUNK). The result is titled "EXSAMP" in the code.

- CPExpert divides the number of CPU Capping samples (R723CCCA) by the EXSAMP value, to yield the percent of execution samples in which the SRM found an address space was CPU capped. The average transaction response time is multiplied by the resulting percentage to yield the amount of time when the average transaction was delayed because of CPU capping.

- Server service classes might serve multiple subsystem service classes. For example, a CICS region (the server) might provide service to a number of service classes which describe different CICS transactions. If the server provides service to multiple service classes, the above

technique automatically "pro-rates" the CPU capping delay of the server. This automatic "pro-rating" is possible because the sampling process of the SRM is independent of the transactions executing.

Resource capping is a way of controlling the distribution of CPU service to one or more service classes. Resource capping is implemented by defining "resource groups" to the Workload Manager. A resource group is simply a named set of two values: a minimum CPU service specification and a maximum CPU service specification. The specifications are in terms of **unweighted CPU service units** (that is, the CPU service coefficients are not applied to TCB nor SRB raw CPU service units).

The Workload Manager will attempt to provide the minimum CPU service to the resource group and will restrict the resource from using more than the maximum CPU service.

Service classes are associated with resource groups; however, a particular service class can be associated with only one resource group[1].

It normally is not advisable to use resource groups. IBM provides the facility solely for special cases, and IBM does not contemplate resource groups being normally used.

Resource group specifications are "preemptive" in nature, in that the Workload Manager attempts to honor resource group specifications before considering other service specifications. Consequently, **resource group specifications could nullify the rest of the Workload Manager's algorithms**.

When the maximum CPU service specified in the resource group has been used, the Workload Manager marks "non-dispatchable" the TCBs and SRBs associated with the service classes assigned to the resource group. This is the situation addressed by Rule WLM222.

As the System Resources Manager takes its samples of the state of address spaces, it examines whether a dispatchable unit (TCB or SRB) is marked non-dispatchable because of a resource group maximum. Samples reflecting the resource group maximum are recorded by RMF in the SMF Type 72 delay samples, as CPU Capping Delay (R723CCCA).

As described earlier, CPExpert computes the percent of CPU Capping Delay for the server service class, as a function of the overall execution of transactions served by the server service class. CPExpert produces Rule

[1]Please see Section 4 (Chapter 1.6) for a discussion of resource groups and how the Workload Manager implements the resource group specifications.

WLM222 if the percent of CPU Capping Delay for the server service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

With Rule WLM222, CPExpert provides the total number of ending transactions in the RMF measurement interval, the total CPU service units required to service the transactions, the average CPU service units per transaction, and the average percent resource capping delay to transactions active in the service class.

Suggestion: As mentioned above, resource groups are intended for very special situations. In most environments, it is far better to allow the Workload Manager to manage system resources to meet the performance goals specified for various service classes. Using resource groups takes control away from the Workload Manager.

Further, specifying maximum CPU service units may result in unused CPU capacity if there are no other service classes ready to use the CPU service.

CPExpert suggests that you consider the following alternatives:

- While there may be unusual situations in which control must be removed from the Workload Manager, please consider whether you have such an unusual situation. If you do not have an unusual situation, you may wish to remove the resource group from the service class. This is particularly true since the service class missing its performance goal describes response goals.

- Alternatively, you should review the performance goal specified for the service class identified by Rule WLM222. CPExpert performs "delay analysis" only on service classes which fail to achieve their performance goal. Consequently, the service class identified by Rule WLM222 had failed to achieve its performance goal.

  The performance goal may be incompatible with the resource group Capacity Maximum, and you may wish to either increase the performance goal (for response goals) or decrease the performance goal (for execution velocity goals).

- Alternatively, you should review the CPU usage report produced by CPExpert at the end of the normal rule listing. Compare the CPU time used by the service class identified by Rule WLM222 with the CPU time used by other service classes. Pay particular attention to CPU time used by any service classes at the same or lower importance, to see whether these service classes should receive the CPU service indicated.

- Alternatively, you may wish to increase the Capacity Maximum specified for the resource group. Since applications executing in the service class are being delayed because of CPU capping, you may remove or decrease the delay by increasing the Capacity Maximum for the resource group.

- Alternatively, you may wish to review the applications executing in the service class identified by Rule WLM222, to determine whether the application code can be optimized so that less CPU time is required.

- Alternatively, you may wish to examine the CICS region parameters to determine whether appropriate specifications have been provided. For example, the System Initialization Table (SIT) parameters often can significantly alter the amount of CPU time required to support CICS transactions.

  If you have licensed the CICS Component of CPExpert, you should execute the CICS Component against the region serving the transactions related to the service class missing its performance goal.

- If none of the above alternatives apply and if Rule WLM222 continually is produced for the service class, you may wish to exclude the service class from CPExpert's analysis. There is little point in having findings produced which cannot be acted upon.[2]

[2]Please see Section 2 for information on how to exclude service classes from analysis.

Rule WLM250: Service Class waited for access to CPU

Finding: CPExpert has determined that waiting for access to a CPU was a major cause of the service class not achieving its performance goal.

Impact: The impact of this finding depends upon the percent of time transactions in the service class were denied access to a CPU. A high percent denied CPU access means HIGH IMPACT while a low percent denied CPU access means LOW IMPACT. See the output associated with the rule which caused this rule to be invoked (Rule WLM101 to Rule WLM103, depending upon the type of service class and performance goal).

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal

Discussion: As the System Resources Manager takes its samples of the state of address spaces, it examines whether a TCB or SRB associated with the address space is waiting for dispatching to a CPU, or whether a TCB is waiting for a local lock.

If an address space is waiting for dispatching, it is being denied access to a CPU because processors are active with higher priority address spaces or with address spaces at the same dispatching priority as the address space waiting for dispatching. Samples reflecting the time address spaces are denied access to a CPU are recorded by RMF in the SMF Type 72 delay samples, as CPU Delay (R723CCDE)[1].

CPExpert computes the percent of CPU Delay for the service class, as a function of the overall execution of transactions executing in the service class. CPExpert produces Rule WLM250 if the percent of CPU Delay for the service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

With Rule WLM250, CPExpert provides the total number of ending transactions in the RMF measurement interval, the total CPU service units

[1]The address space could also be waiting for dispatch because the Workload Manager has marked the TCB or SRB "non-dispatchable" because of CPU Capping. Please see Section 4 (Chapter 1.6) for a discussion of resource groups and how the Workload Manager implements the resource group specifications. The CPU Delay samples recorded in R723CCDE do **not** include any samples of waiting because of CPU Capping. CPU Capping Delay is recorded in a separate SMF Type 72 variable (R723CCCA).

©Copyright 1994, Computer Management Sciences, Inc.

consumed by the service class, and the percent of active time when transactions in the service class were denied access to a CPUs. Additionally, CPExpert provides summary information about the CPU time used by service classes with higher importance, the same importance, and lower importance with respect to the service class failing to achieve its performance goal.

The CPU time used by other levels of importance can be used in association with the CPU USED by the service class missing its performance goal, to assess whether the problem is caused by service classes with a higher importance or service classes at the same level of importance.

The following example illustrates the output from Rule WLM250:

```
RULE WLM250: SERVICE CLASS WAITED FOR ACCESS TO CPU

   Service Class TSO (Period 1) was delayed waiting for CPU
   dispatching.  During the following RMF measurement intervals, a TCB
   or SRB was waiting to be dispatched, or a TCB was waiting for a local
   lock.  The "% Denied CPU" value represents the percent of TSO's
   active time when TSO was waiting for access to a CPU.  CPExpert
   will produce a report at the end of this analysis which shows the CPU
   time used by all service class periods.

                                          %       CPU TIME USED BY OTHER
                              CPU USED   DENIED   ---LEVELS OF IMPORTANCE---
   MEASUREMENT INTERVAL    (TSO -- 1)    CPU    HIGHER      SAME     LOWER
   13:02-13:07,21JUN1994     0:02:10    31.5   0:00:43   0:05:31   0:00:00
   13:07-13:12,21JUN1994     0:02:09    29.6   0:00:51   0:05:30   0:02:14
   13:17-13:22,21JUN1994     0:02:14    50.9   0:00:49   0:05:42   0:02:09
   13:22-13:27,21JUN1994     0:02:09    35.9   0:00:45   0:05:25   0:02:10
```

Please note that CPExpert does not produce Rule WLM250 for "served" service classes (e.g., a service class describing CICS transactions). The SRM does not collect resource information for "served" service classes. Rather, the SRM collects resource information at the "server" service class level (e.g., at the CICS region). CPExpert will analyze the "server" service class to identify constraints and Rule WLM255 may result from this analysis.

Suggestion: When a service class fails to achieve its goal because it is denied access to a CPU, you have several alternatives:

- **Increase the importance of the service class**. The Workload Manager attempts to achieve the performance goal for each service class. When the Workload Manager detects that a service class is not achieving its performance goal, the Workload Manager will assess whether changing

the existing distribution of system resources will help a service class achieve its performance goal[2].

The Workload Manager examines (and attempts to help) service classes in descending order of importance. Importance levels may be specified as values of 1 to 5, with Importance 1 being the most important and Importance 5 being the least important. Importance 0 is an implied importance level for system tasks, and Importance 6 is an implied importance level for service classes with a Discretionary performance goal.

If you increase the importance of a service class, the Workload Manager will give a higher priority to the service class when resources are allocated. Of particular relevance to the problem of a service class being denied access to a CPU is that the Workload Manager may assign a higher dispatching priority to address spaces in the service class if the service class is missing its goal. With a higher dispatching priority, the service class will be less likely to be denied access to a CPU.

- **Decrease the importance of another service class**. The Workload Manager will attempt to provide resources to help service classes missing their performance goal. As described above, the Workload Manager examines (and attempts to help) service classes in descending order of importance.

  You should examine the importance specified for (1) service classes with a higher importance and (2) service classes at the same importance as the service class missing its performance goal. Determine whether these importance levels match the management objectives of your installation.

- **Alter the performance goal specified for the service class**. You should assess whether the performance goal is appropriate for the applications assigned to the service class. Perhaps the performance achieved is adequate, or perhaps the specified performance goal can be altered so that the service class meets its objective at the existing level of service. That is, the delivered service may be adequate for management objectives and you may need to change the performance goal specified to the Workload Manager.

- **Alter the performance goal specified for another service class**. You should assess whether the performance goal is appropriate for the applications assigned to other service classes. The Workload Manager attempts to achieve the performance goal for each service class. When the Workload Manager detects that a service class is not achieving its

[2]Please refer to Section 4 for a more comprehensive discussion of the Workload Manager's algorithms.

performance goal, the Workload Manager will assess whether changing the existing distribution of system resources will help a service class achieve its performance goal.

As described above, the Workload Manager first examines service classes based on importance. However, if several service classes are of the *same* importance, the Workload Manager will attempt to help the service class having the *worst* performance (as measured by the performance index).

You should assess whether appropriate performance goals have been specified for other service classes at a higher importance or at the same importance.

- **Reschedule workloads**. Your organization may be able to reschedule conflicting workloads to another system to eliminate the conflicts for processor access.

- **Add another processor**. You may be able to add another processor (potentially not so difficult in an LPAR environment). Adding another processor will provide another "CPU server" from a queuing model view; having another "CPU server" significantly reduces the probability that an address space will be denied access to a CPU[3].

- **Acquire faster processors**. If the service class missing its performance goal is sufficiently important and it is being denied access to a CPU, you may be able to solve the problem by acquiring faster processors.

- **Ignore the finding**. There may be situations in which you wish to simply ignore CPExpert's finding. You might not care that a low priority batch service class is denied access to the CPU. If this is the case, perhaps you should not have a performance goal associated with the workload. However, you may wish to have a performance goal (and have CPExpert perform analysis) simply to assess other delays. For example, you may wish to assess the auxiliary paging delays experienced by the workload.

  Another (and potentially more common) reason a service class period is denied access to a CPU is caused by the inherent processing characteristics of the workload, along with the MVS dispatching algorithms. Please refer to Rule WLM251 for a discussion of this situation. Rule WLM251 will be produced if CPExpert believes that the service class period is denied access to a CPU because of this situation.

[3]Please refer to *Probability, Statistics, and Queuing Theory* by Arnold O. Allen for a description of the M/M/C queuing model which can be used to assess the effect of changing the number of processors.

- **Exclude the service class from analysis**. If none of the above alternatives apply and if Rule WLM250 continually produces for the service class, you may wish to exclude the service class from CPExpert's analysis. There is little point in having findings produced which cannot be acted upon. Please see Section 3 (Chapter 1.1.8) for information on how to exclude service classes from analysis.

After CPExpert has completed its analysis of performance constraints, a summary of CPU time used by each service class period is produced for any measurement interval in which a service class did not achieve its performance goal and the service class was significantly denied access to a processor.

The following example illustrates the report which is produced:

```
      SUMMARY OF SERVICE CLASS CPU TIME CAPTURED IN TYPE 72 RECORDS

                        SERVICE  CLASS                 GOAL                 %
 MEASUREMENT INTERVAL    CLASS  PERIOD  GOAL TYPE     IMPORT  CPU USED     CPU

  21JUN1994:13:07:01   SYSSTC     1    SYSTEM TASKS     0    0:00:36.29   6.6
  21JUN1994:13:07:01   SYSTEM     1    SYSTEM TASKS     0    0:00:14.57   2.6
  21JUN1994:13:07:01   CICSRGN    1    SERVER CLASS     2    0:02:00.11  21.8
  21JUN1994:13:07:01   IMSCTL     1    SERVER CLASS     2    0:00:37.87   6.9
  21JUN1994:13:07:01   IMSMP      1    SERVER CLASS     2    0:01:18.97  14.3
  21JUN1994:13:07:01   TSO        1    AVG RESPONSE     2    0:01:08.24  12.4
  21JUN1994:13:07:01   TSO        2    AVG RESPONSE     2    0:00:19.16   3.5
  21JUN1994:13:07:01   TSO        3    AVG RESPONSE     2    0:00:42.02   7.6 DENIED CPU(67%)
  21JUN1994:13:07:01   BATCHHI    1    EX. VELOCITY     3    0:02:02.73  22.3
  21JUN1994:13:07:01   BATCHLOW   1    EX. VELOCITY     3    0:00:11.40   2.1

  TOTAL SERVICE CLASS CPU TIME CAPTURED IN TYPE 72 RECORDS:  0:09:22.75
```

The CPU USED column reflects the total TCB and SRB CPU time used by the service class during the measurement interval. The "% CPU USED" reflects the percent of "TOTAL SERVICE CLASS CPU TIME CAPTURED IN TYPE 72 RECORDS" which was used by the service class.

Not all CPU time is accounted for by MVS. As much as 25% CPU time has been documented in the literature as "unrecovered" CPU time - the CPU time which is not included in TCB or SRB CPU time recorded by SMF in Type 72 records. Consequently, the "TOTAL SERVICE CLASS CPU TIME CAPTURED" may be significantly less than the CPU time actually used by service classes.

CPExpert annotates any service class which was denied access to the CPU as a primary or secondary cause of the service class failing to achieve its performance goal. Along with the annotation, CPExpert shows the

percent of service class active time when an address space was denied access to a processor.

This report will allow you to assess the CPU time used by different service classes, by level of importance. To facilitate this review, the service class information is ordered by Importance associated with each service class.

Please note that the distribution of CPU time may include CPU time associated with SERVER service classes. The goal importance of the SERVER service classes is ignored after address space start-up. The importance of the SERVER service classes is a function of the service classes being served. Consequently, the CPU times may be misleading, as the CPU times shown for SERVER service classes may be at a higher or lower importance than that defined for the SERVER service class.

CPExpert identifies the **highest** goal importance of any served service class, and displays this highest goal importance for the server service class. **This goal importance may be different from the goal importance which was defined for the server service class using the Workload Manager ISPF panel.**

No information is available to identify the CPU time used by the server to support different served service classes. Consequently, if the served service classes have different goal importance, you may be unable to determine whether the distribution of CPU time properly reflects what was actually required to support different goal importance levels. On the other hand, if the served service classes have the same goal importance, then the report properly reflects the CPU time used at the specified goal importance level.

Rule WLM251: Reduced Preemption may have caused service class CPU delay

Finding: CPExpert believes that the MVS reduced preemption algorithms may have caused the service class to experience CPU delay.

Impact: The impact of this finding depends upon whether CPExpert's assessment of the cause of CPU delay is correct. If the reduced preemption algorithms did cause CPU delay, this finding is produced primarily for information purposes.

Logic flow: The following rules cause this rule to be invoked:
Rule WLM250: Service Class waited for access to CPU

Discussion: As the System Resources Manager takes its samples of the state of address spaces, it examines whether a TCB or SRB associated with the address space is waiting for dispatching to a CPU, or whether a TCB is waiting for a local lock.

If an address space is waiting for dispatching, it is being denied access to a CPU because processors are active with higher priority address spaces or with address spaces at the same dispatching priority as the address space waiting for dispatching. Samples reflecting the time address spaces are denied access to a CPU are recorded by RMF in the SMF Type 72 delay samples, as CPU Delay (R723CCDE)[1].

Another reason a service class period can be denied access to a CPU is due to the inherent processing characteristics of the workload, along with the MVS dispatching algorithms.

- Dispatchable units (address spaces and enclaves) in the service class period may use the CPU in short bursts. That is, they execute for a short time and then relinquish control of the processor.

- If a higher priority dispatchable unit immediately interrupts an executing dispatchable unit, processor internal high-speed cache must be purged and reloaded. This process defeats some of the hardware design performance of larger systems. IBM studies showed

[1]The address space could also be waiting for dispatch because the Workload Manager has marked the TCB or SRB "non-dispatchable" because of CPU Capping. Please see Section 4 (Chapter 1.6) for a discussion of resource groups and how the Workload Manager implements the resource group specifications. The CPU Delay samples recorded in R723CCDE do **not** include any samples of waiting because of CPU Capping. CPU Capping Delay is recorded in a separate SMF Type 72 variable (R723CCCA).

©Copyright 1997, Computer Management Sciences, Inc.

that it may be better to allow the lower priority dispatchable unit to continue executing for a short time, in hopes that it would voluntarily release control.

Based on these IBM studies, the *reduced preemption* algorithms were implemented in MVS/ESA SP3.1. Successive releases of MVS have improved the algorithms, but the basic concept remains. With reduced preemption, a lower priority dispatchable unit is not necessarily interrupted immediately when a higher priority dispatchable unit becomes ready to execute. Rather, the dispatchable unit usually is allowed to continue executing for a short time (a few milliseconds). MVS monitors how well the algorithm works (on a dispatchable unit-by-dispatchable unit basis) and modifies the reduced preemption as necessary.

- If a high priority dispatchable unit executes for only a short time, the amount of time it is delayed by the reduced preemption algorithms could be large relative to the time spent executing.

- Consider that execution velocity (for example) is based on CPU Using divided by (CPU Using, plus Delay for CPU or processor storage)[2]. Suppose that a particular task uses only 1 millisecond of CPU when it is dispatched and the reduced preemption algorithm delays execution for 3 milliseconds.

  The <u>best</u> execution velocity that could be achieved by this task under these conditions would be 25 (1 millisecond / (1 millisecond + 3 milliseconds). Even though you might have specified an execution velocity goal of 90 for the task, you could never achieve the specified goal. This effect is startling and counter-intuitive.

As shown by the above discussion, it is possible that a service class period may miss its performance goal because it is denied access to a CPU, and there might be no action that can be taken to provide better access. Neither increasing the velocity goal nor specifying a higher importance will have any effect in this situation. The "missing goal" status is caused by the processing characteristics of address spaces in the service class period, matched with the MVS Dispatcher algorithms.

CPExpert attempts to gain some insight into the likelihood of this situation occurring. CPExpert produces Rule WLM251 when it observes that the following conditions were present in the data presented by Rule WLM250, for a significant percent of the RMF intervals:

[2] I/O Using and I/O Delays optionally may be included in this algorithm beginning with OS/390 Release 3.

- A small amount of CPU resources were used by the service class period.

- The CPU delay was much higher than would be expected based on the CPU time used by service class periods at a higher or same level of importance. CPExpert applies a queuing model to estimate the CPU delay that would be experienced based on the CPU time used by service classes at a higher importance and at the same level of importance as the service class denied CPU. The result of the model (multiplied by a factor of two[3]) is compared with the actual delay experienced.

- A relatively large amount of CPU resources were used by service class periods at a lower importance.

When these three conditions are present in the data, CPExpert believes it is likely that the performance goal was missed because of inherent characteristics of the applications and the dispatcher algorithms.

The following example illustrates the sequence of CPExpert findings what lead to Rule WLM251.

- In the example output, the APPCFEED service class period had an execution velocity goal of 50.

- As reported by Rule WLM103, this service class period missed its performance goal. The primary cause of delay was DENIED CPU, which caused 100% of the delay.

- Rule WLM250 expanded on this analysis, reporting that the APPCFEED service class used a minuscule amount of CPU resources, while service class periods at the same or lower levels of goal importance used a significant amount of CPU.

  Please note that there is not a direct relationship between goal importance and dispatching priority. The Workload Manager adjusts dispatching priority based on whether CPU use is a constraint and it is possible that a service class period with a lower goal importance will have a higher dispatching priority than one with a higher goal importance.

  However, once a service class period is missing its goal and the Workload Manager detects that it is being denied access to CPU resources, it is unlikely that lower importance work would have a higher dispatching priority! Since the service class period (1) did miss its performance goal and (2) being denied CPU access was the major

[3]The multiplier is used to prevent spurious findings.

reason for missing its goal, it is unlikely that the lower importance work was assigned a higher dispatching priority.

- Since there was significant CPU use at a lower importance and very small CPU use by the APPCFEED service class period, CPExpert concludes that APPCFEED probably missed its goal because of reduced preemption. Rule WLM251 reports this conclusion.

```
RULE WLM103: SERVICE CLASS DID NOT ACHIEVE VELOCITY GOAL

   APPCFEED (Period 1): Service class did not achieve its velocity goal
   during the measurement intervals shown below.  The velocity goal was
   50% execution velocity, with an importance level of 2.  The '% USING'
   and '%TOTAL DELAY' percentages are computed as a function of the average
   address space ACTIVE time.  The 'PRIMARY,SECONDARY CAUSES OF DELAY'
   are computed as a function of the execution delay samples on the local
   system.

                         ------LOCAL SYSTEM--------
                           %    % TOTAL EXEC   PERF  PLEX PRIMARY,SECONDARY
   MEASUREMENT INTERVAL  USING   DELAY  VELOC  INDX   PI  CAUSES OF DELAY
   14:45-15:00,01MAR1994   5.7   46.3    11%   4.55  4.55 DENIED CPU(100%)


RULE WLM250: SERVICE CLASS WAITED FOR ACCESS TO CPU

   APPCFEED (Period 1): Service class was delayed waiting for access to
   a CPU.  During the following RMF measurement intervals, a TCB or
   SRB was waiting to be dispatched, or a TCB was waiting for a local
   lock.  The "% DENIED CPU" value represents the percent of APPCFEED's
   EXECUTING time when APPCFEED was waiting for access to a CPU.  CPExpert
   will produce a report at the end of this analysis which shows the CPU
   time used by all service class periods.

                                          %       CPU TIME USED BY OTHER
                              CPU USED   DENIED   ---LEVELS OF IMPORTANCE---
   MEASUREMENT INTERVAL   APPCFEED-1    CPU    HIGHER      SAME     LOWER
   14:45-15:00,01MAR1994    0:00:01    46.3   0:15:19   0:32:29   0:19:19


RULE WLM251: CPU DELAY MAY BE CAUSED BY REDUCED PREEMPTION

   APPCFEED (Period 1): Service class period was delayed waiting for
   access to a CPU, as described in Rule WLM250.  However, for 100% of the
   RMF measurement intervals shown in Rule WLM250, the service class used
   very little CPU, the CPU delay was much more than would be expected
   considering the CPU used by service class periods at a higher or same
   importance, and service class periods at a lower importance used a
   significant amount of CPU.  These conditions lead CPExpert to believe
   that perhaps the reduced preemption algorithms were responsible for the
   service class being denied access to a CPU.  You can assess whether this
   is a likely reason the service class period was denied access to a CPU
   by reviewing the information presented with Rule WLM250 and by reviewing
   the CPU usage reports produced at the end of CPExpert's analysis (along
   with your knowledge of the type of work assigned to the service class
   period).
```

Suggestion: CPExpert suggests that you examine the work assigned to the service class period identified by this finding. Typically, the work will be started tasks that have short bursts of CPU use.

If CPExpert's conclusion about the processing nature of the work is correct, there may not be any way to prevent the service class period from missing its performance goal, so long as you have assigned the work to a service class having a specified performance goal. The delays inherent in the MVS reduced preemption algorithms may not permit the goal to be attained.

CPExpert suggests that you consider the following alternatives:

- **Reassess the need for the service class period**. You may wish to examine the work assigned to the service class period, and determine that there is no need to define a separate service class period for the particular work units. You may be able to assign the work to a different service class period and eliminate the existing service class period. This action would reduce system overhead.

  IBM SRM/WLM developers have indicated that a small number of service class periods is desirable. They have observed that the Workload Manager algorithms typically become increasingly ineffective when a site has specified a large number of service class periods.

- **Assign the work to SYSSTC service class**. You should assess the importance of the work assigned to the service class period. If the work is sufficiently important, and if the amount of CPU resources is very low, you may wish to assign the work to the SYSSTC service class. Work assigned to the SYSSTC system service class are outside the normal dispatching priority management controlled by the Workload Manager[4].

- **Ignore the finding**. You may wish to simply ignore CPExpert's finding. However, you might want to leave the work assigned to a service class period and specify a performance goal (and have CPExpert perform analysis) simply to assess other delays. For example, you may wish to assess the auxiliary paging delays experienced by the workload.

- **Exclude the service class from analysis**. If none of the above alternatives apply and if Rule WLM250 and Rule WLM251 continually be produced for the service class, you may wish to exclude the service class from CPExpert's analysis. There is little point in having findings produced which cannot be acted upon. Please see Section 3 (Chapter 1.1.8) for information on how to exclude service classes from analysis.

Reference: MVS Planning: Workload Management
MVS/ESA(SP 5): Chapter 8: Defining Service Classes and Performance Goals

[4]Address spaces in SYSSTC service class execute at dispatching priority FD (253) if APAR OW19265 is **not** applied, and execute at dispatching priority of FE (254) if OW19265 **is** applied.

OS/390 (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R4):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R5):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R6):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R7):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R8):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R9):	Chapter 8: Defining Service Classes and Performance Goals
OS/390 (V2R10):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R1):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R2):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R3):	Chapter 8: Defining Service Classes and Performance Goals
z/OS (V1R4):	Chapter 8: Defining Service Classes and Performance Goals

"MVS Workload Manager Velocity Goals: What you don't know can hurt you", John Arwe, IBM Corporation, CMG'96 Proceedings.

"MVS/ESA Full vs. Reduced/Partial Preemption", Steve Lamborne, Hitachi Data Systems Corporation, CMG'94 Proceedings.

Rule WLM252: CPU access might be denied because of Resource Group minimum

Finding: CPExpert believes that CPU access might have been denied for the service class missing its performance goal because some other service class was assigned to a Resource Group with a **minimum CPU service** specification.

Impact: This finding should be viewed as generally having a HIGH IMPACT on the performance of your computer system.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM250: Service class waited for access to CPU

Discussion: When CPExpert determines that a service class waited for access to a CPU, CPExpert continues to analyze the data trying to identify why the service class was denied access.

From a simplistic view, the service class was denied access because address spaces with a higher dispatching priority used the CPU. With SP5 (Goal Mode), user do not assign dispatching priority to workloads. The Workload Manager assigns dispatching priority based on user performance goals and goal importance for different service classes, and based on how well the service classes meet their performance goals.

Additionally, users can assign service classes to resource groups, and this assignment can cause the Workload Manager to grant or deny access to a CPU by address spaces.

A resource group is simply a "named" description of the total minimum and maximum **unweighted** CPU service units per second which may be used by one or more service classes assigned to the resource group. A resource group is defined using the *Create a Resource Group* panel in the Workload Manager ISPF application. A resource group applies across an entire sysplex. Service classes[1] are assigned to a resource group using the *Create a Service Class* panel in the Workload Manager ISPF application.

[1]A resource group may not be associated with a service class representing subsystem transactions (i.e., a service class defined for transactions executing under CICS/ESA Version 4.1 or under IMS/ESA Version 5). This is because CPU resources are not monitored by the SRM for the transactions; the CPU resources are monitored at the **address space level** (e.g., the CICS region or IMS message processing region). Further, CPU dispatching occurs at the address space level, rather than at the transaction level. Since CPU usage is not collected at the transaction level and CPU dispatching is at the address space level, the Workload Manager cannot control the amount of CPU resources allocated to service classes which represent transactions.

©Copyright 1994, Computer Management Sciences, Inc.

The Workload Manager will attempt to provide the specified minimum CPU service to the resource group. The Workload Manager attempts to provide the specified minimum CPU service to the resource group by adjusting the dispatching priority of service classes assigned to the resource group. The Workload Manager will restrict service classes assigned to the resource group from using more than the specified maximum CPU service. The Workload Manager uses "CPU capping" to restrict the total amount of CPU service used by service classes assigned to the resource group.

There are potentially serious effects of specifying a minimum CPU service for a resource group. The effect is caused by the order in which the Workload Manager selects service classes for policy adjustment.

- The Workload Manager first determines whether any resource group is below the **minimum** CPU service specification. If the minimum CPU specification is not being provided, the Workload Manager takes the following actions in an attempt to provide the minimum CPU service:

  - The Workload Manager determines whether any service class assigned to the resource group is not meeting its performance goal. If any service class is not meeting its performance goal, the Workload Manager increases the dispatching priority (if appropriate) of the service class.

  - If no service classes assigned to the resource group were missing their performance goal, the Workload Manager increases the dispatching priority (if appropriate) of all service classes assigned to the resource group. The dispatching priority of all service classes assigned to the resource group (including those service classes with a discretionary goal[2]) may be increased.

- After the Workload Manager performs the above tasks, the Workload Manager may examine service classes based on the Goal Importance of the service classes.

The result of the above process can be that service classes with a low importance (or even service classes with a discretionary goal) can be assigned CPU dispatching priority above that which is assigned to the service classes with the highest Goal Importance! The resulting CPU dispatching priorities and CPU demands can result in service classes with high Goal Importance missing their performance goals.

---

[2]Please note that the *MVS/ESA SP5 Planning: Workload Management* document is incorrect. This document states in the *Defining Resource Groups* section that "If there is a resource group defined for a service class with a discretionary goal, workload management achieves the minimum as long as the goals of work running in any other service class are not impacted. If other performance goals are impacted, then workload management does not maintain the minimum." Based on personal communication with the Workload Manager developer who wrote the specific code which attempts to provide the minimum specified CPU service, these statements are incorrect in the IBM document and the description provided above is what actually transpires.

This may not be the effect you wish, but the Workload Manager simply follows the specific direction provided for the resource group, namely, that a minimum CPU service was specified for the resource group and this minimum should be provided.

When a service class missed its performance goal and the service class was denied access to a CPU, CPExpert determines whether any service classes were assigned to resource groups with a **minimum CPU** specification. If so, CPExpert computes the CPU service used by service classes which were assigned to each resource group with a minimum CPU specification. The computations are done separately for service classes at a lower goal importance or at the same goal importance.

The purpose of the computations is to estimate whether resource group minimum CPU specifications may have caused the service class to be denied access. The result is simply an estimate of the potential impact; the SMF Type 72 records do not contain dispatching priority for service classes (the dispatching priority is dynamically adjusted by the Workload Manager).

CPExpert produces Rule WLM252 if any service classes were assigned to a resource group with a minimum CPU specification and these service classes actually used CPU service.

Suggestion: CPExpert suggests that you verify the minimum CPU service specification for resource groups defined in the service policy. Unless there are unique requirements for the minimum CPU service specifications, CPExpert suggests that the minimum be changed to zero.

Reference: MVS Planning: Workload Management

MVS/ESA(SP 5):	Chapter 7: Defining Resource Groups
OS/390 (V1R1):	Chapter 7: Defining Resource Groups
OS/390 (V1R2):	Chapter 7: Defining Resource Groups
OS/390 (V1R3):	Chapter 7: Defining Resource Groups
OS/390 (V2R4):	Chapter 7: Defining Resource Groups
OS/390 (V2R5):	Chapter 7: Defining Resource Groups
OS/390 (V2R6):	Chapter 7: Defining Resource Groups
OS/390 (V2R7):	Chapter 7: Defining Resource Groups
OS/390 (V2R8):	Chapter 7: Defining Resource Groups
OS/390 (V2R9):	Chapter 7: Defining Resource Groups
OS/390 (V2R10):	Chapter 7: Defining Resource Groups
z/OS (V1R1):	Chapter 7: Defining Resource Groups
z/OS (V1R2):	Chapter 7: Defining Resource Groups
z/OS (V1R3):	Chapter 7: Defining Resource Groups
z/OS (V1R4):	Chapter 7: Defining Resource Groups

Rule WLM255: Service class was in Active state but server was denied access to CPU

Finding: CPExpert has determined that the transaction service class which missed its performance goal was in Active state, but the server service class was denied access to a CPU.

Impact: This finding has a HIGH IMPACT on performance of your computer system.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120(series) and Rule WLM130(series) describe the results from this analysis.

After analyzing the subsystem transaction delays, CPExpert identifies the service classes which serve the transactions. The subsystem transactions typically are CICS transactions, and the servers are the CICS regions. Alternatively, the transactions could be IMS transactions and the servers could be the IMS control regions or transaction processing regions.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records[1].

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS.

[1]Please see Section 4 for a discussion of these states and the sampling process.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert analyzes the CPU requirements of the server service class to determine if the server was denied access to a CPU.

As the System Resources Manager takes its samples of the state of address spaces, it examines whether a TCB or SRB associated with the address space is waiting for dispatching to a CPU, or whether a TCB is waiting for a local lock.

If an address space is waiting for dispatching, it is being denied access to a CPU because processors are active with higher priority address spaces or with address spaces at the same dispatching priority as the address space waiting for dispatching. Samples reflecting the time address spaces are denied access to a CPU are recorded by RMF in the SMF Type 72 delay samples, as CPU Delay (R723CCDE).

For a service class consisting of CICS or IMS regions, the CICS or IMS regions compete with each other at the same dispatching priority. This is because the Workload Manager will create an internal service class ($SRMSnnn) consisting of all address spaces which provide service to the subsystem transactions served by the CICS or IMS service class. The Workload Manager will manage these address spaces collectively from a CPU dispatching view (that is, all address spaces will have the same CPU dispatching priority)[2].

CPExpert computes the percent of CPU Delay for the server service class, as a function of the response time of the subsystem transaction service class missing its performance goal. CPExpert produces Rule WLM255 if the percent of CPU Delay for the subsystem transaction service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

With Rule WLM255, CPExpert provides the total CPU service units consumed by the service class in the RMF measurement interval, the percent of active time when transactions in the service class were denied access to a CPU, and the average multiprogramming level of the server service class. The average multiprogramming level is provided so you can assess whether address spaces might have competed with each other.

The following example illustrates the output from Rule WLM255:

[2]The address spaces will be managed **individually** from a processor storage access policy view.

```
RULE WLM255:  SERVICE CLASS WAS ACTIVE BUT SERVER WAS DENIED CPU

   During the above measurement intervals, the CICUSRTX Service Class was
   in the READY STATE during a significant portion of its response time.
   However, at least one address space in the CICSRGN server was denied
   access to a CPU for a significant percent of this time.  During the
   following RMF measurement intervals, CICSRGN had a TCB or SRB waiting
   to be dispatched, or a TCB was waiting for a local lock.  The below
   information shows the CPU used by CICSRGN during the measurement
   interval, and the "PERCENT DENIED CPU" value represents the percent
   of CICSRGN's ACTIVE time when at least one address space was waiting
   for access to a CPU.  CPExpert will produce a report at the end of this
   analysis which shows the CPU time used by all service class periods.

                                                                AVERAGE
                                 CPU  USED      PCT SERVER     SERVER
   MEASUREMENT INTERVAL          BY SERVER      DENIED CPU       MPL
   13:07-13:12,21JUN1994          0:02:00          60.4          4.0
   13:17-13:22,21JUN1994          0:02:00          57.7          4.0
```

Suggestion; Please refer to the suggestions in Rule WLM250 for a discussion of the alternatives which can be implemented to improve access to a CPU.

Rule WLM256: Service class was in Active state and server was not denied access to CPU

Finding: CPExpert has determined that the transaction service class which missed its performance goal was in Active state, and the server service class was not denied access to a CPU.

Impact: This finding is provided for information purposes only.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM120: Significant transaction time was in Active state

Discussion: When CPExpert produces Rule WLM104 or Rule WLM105 to indicate that a subsystem service class did not achieve its performance goal, the logic of these rules tries to identify the cause of the delay. The cause of the delay initially is analyzed from the "served" service class view. Rule WLM120(series) and Rule WLM130(series) describe the results from this analysis.

Address spaces executing in the system can be in a variety of states from the perspective of the Workload Manager: using the CPU, delayed for an identifiable reason, or delayed for some unknown reason.

The System Resources Manager (SRM) periodically samples the state of each address space in each service class. These samples are accumulated into variables which are recorded by RMF in the "Service Class Period Data Section" of SMF Type 72 (Subtype 3) records.

CPExpert produces Rule WLM120 when a significant cause of delay to a subsystem transaction was that the transaction was in Active state. The Active state indicates that a task was executing on behalf of the transaction, from the perspective of CICS or IMS. CPExpert analyzes the CPU requirements of the server service class to determine if the server was denied access to a CPU.

As the System Resources Manager takes its samples of the state of address spaces, it examines whether a TCB or SRB associated with the address space is waiting for dispatching to a CPU, or whether a TCB is waiting for a local lock.

If an address space is waiting for dispatching, it is being denied access to a CPU because processors are active with higher priority address spaces

©Copyright 1994, Computer Management Sciences, Inc.

or with address spaces at the same dispatching priority as the address space waiting for dispatching. Samples reflecting the time address spaces are denied access to a CPU are recorded by RMF in the SMF Type 72 delay samples, as CPU Delay (R723CCDE).

CPExpert computes the percent of CPU Delay for the server service class, as a function of the response time of the subsystem transaction service class missing its performance goal. CPExpert produces Rule WLM255 if the percent of CPU Delay for the subsystem transaction service class is greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

CPExpert produces Rule WLM256 if the percent of CPU Delay for the subsystem transaction service class is not greater than the significance value specified in the **WLMSIG** guidance variable in USOURCE(WLMGUIDE). The finding means that CPU time actually used by tasks processing the subsystem service class transactions accounted for a significant amount of the response time of these transactions. These tasks were not normally preempted from the using a CPU by higher priority processing.

The following example illustrates the output from Rule WLM256:

```
RULE WLM256: SERVICE CLASS WAS ACTIVE AND SERVER WAS NOT DENIED CPU

  During some of the above measurement intervals, the CICUSRTX Service
  Class was in the ACTIVE STATE during a significant portion of its
  response time and the CICSRGN server was not denied access to a CPU
  for any significant amount of time during these intervals.  This
  finding means that CPU time actually used by tasks processing CICUSRTX
  transactions accounted for a significant amount of the response time
  of these transactions, and these tasks were not normally preempted from
  the using a CPU by higher priority processing.
```

Suggestion; Please refer to Rule WLM212 for a discussion of alternatives.

Rule WLM340: Batch jobs may be delayed waiting for an initiator

Finding: CPExpert believes that much of the UNKNOWN delay may be attributed to batch jobs waiting for an initiator.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class. The amount of impact depends upon the amount of delay attributed to waiting for an initiator.

NOTE: This finding applies only to environments prior to OS/390 V2R4. With OS/390 V2R4, batch job classes may be managed by the Workload Manager, and other CPExpert analysis applies.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM300: Service Class was delayed for UNKNOWN delay

Discussion: When CPExpert detects that a service class did not achieve its response goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rule).

When the UNKNOWN delay is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE), CPExpert analyzes several possible causes of delay outside the control of the SRM. One of the possible causes of UNKNOWN delay for a service class describing batch workload is that the batch jobs were delayed because they were waiting for an initiator.

With RMF Version 5, SMF Type 72 (Subtype 3) records contain the total transaction elapsed time in field R723CTET[1]. The transaction elapsed time is measured from the point of entry to the point of termination of the transaction. This time includes both queued time and active time. For a batch job, the queued time represents the time that the batch job was on a JES queue waiting for an initiator. For an APPC transaction, the queue time represents the time that the APPC transaction waited for the APPC/MVS transaction scheduler.

With RMF Version 5, SMF Type 72 (Subtype 3) records contain the total transaction execution time in field R723CXET. For batch jobs, the transaction execution time represents the time that a batch job had been started by an initiator.

[1]The meaning of the R723CTET field in Goal Mode is the same as the SMF72TST field in versions provided in SMF Type 72 (Subtype 1) for Compatibility Mode and for earlier versions of MVS.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert computes the average amount of time transactions spent in a queue by subtracting the transaction execution time (R723CXET) from the total transaction elapsed time (R723CTET). CPExpert concludes that queue delay time was a significant amount of the transaction time if the queue delay time divided by the total transaction time is greater than the **WLMSIG** guidance variable.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "batch" and assumes that the service class describes batch workload if "batch" is encountered. CPExpert produces Rule WLM340 if "batch" is detected in the Service Class Description.

It is, of course, possible that the service class does not describe batch workload even though "batch" is in the description. This instance is unlikely, as most installations will use the word "batch" in a description of only batch work.

It also is possible that the word "batch" will not be in the description of a service class of batch workload. Rule WLM341 will be invoked to provide information in this case.

The following example illustrates the output from Rule WLM340:

```
RULE WLM340: BATCH JOBS MAY BE DELAYED WAITING FOR AN INITIATOR

  The HOTBATCH Service Class might have failed to achieve its performance
  goal because batch jobs were waiting for an initiator.  The below
  information shows the average number of address spaces in the system,
  by category.
                          -------THIS SERVICE CLASS-------       AVERAGE
                          AVERAGE QUEUE    AVERAGE     AVG        BATCH
  MEASUREMENT INTERVAL    TIME PER JOB   JOBS QUEUED   MPL     INITIATORS
  13:02-13:07,21JUN1994     0:05:02           2        0.1          16
```

Suggestion: CPExpert suggests that you review your initiator structure. Depending upon the amount of delay and the importance of the batch workload, you may wish to revise the initiator structure (provide more initiators, change the classes assigned to initiators, etc.).

Alternatively, you may wish to revise the job class assigned to the batch workload missing the performance goal.

From a practical matter, Rule WLM340 normally will be produced only for a service class consisting of test batch jobs. This is because you are unlikely to assign a response goal to lengthy batch jobs. (In fact, CPExpert will detect a lengthy response goal and produce Rule WLM006. Please

refer to the documentation of Rule WLM006 for a discussion of the implications of a lengthy response goal.)

Batch work with an execution velocity goal cannot miss its performance goal because of initiator delays. This is because such delays are not included in the definition of execution velocity[2]. This batch work can, of course, exhibit poor performance because of initiator delays. This poor performance will not be detected by the Workload Manager, and consequently will not be detected by CPExpert.

[2]Queue delay (including time spent waiting for an initiator) optionally will be included in execution velocity beginning with OS/390 Version 2 Release 4.

Rule WLM341: Work may be delayed waiting for initiation or scheduling

Finding: CPExpert believes that much of the UNKNOWN delay may be attributed to work waiting for a initiation or scheduling.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class. The amount of impact depends upon the amount of delay attributed to waiting for initiation or scheduling.

NOTE: This finding applies only to environments prior to OS/390 V2R4. With OS/390 V2R4, batch job classes may be managed by the Workload Manager, and other CPExpert analysis applies.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300: Service Class was delayed for UNKNOWN delay
Rule WLM340: Batch jobs may be delayed waiting for an initiator

Discussion: When CPExpert detects that a service class did not achieve its response goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules).

In Rule WLM340, CPExpert computes the amount of time spent in a queue by subtracting the transaction execution time (R723CXET) from the total transaction elapsed time (R723CTET). CPExpert concludes that queue delay time was a significant amount of the transaction time if the queue delay time divided by the total transaction time is greater than the **WLMSIG** guidance variable.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "batch" and assumes that the service class describes batch workload if "batch" is encountered. CPExpert produces Rule WLM340 if "batch" is detected in the Service Class Description.
It is possible that the word "batch" will not be in the description of a service class of batch workload, or the workload may describe APPC transactions. Rule WLM341 will be invoked to provide information in this case.

The following example illustrates the output from Rule WLM341:

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM341: SERVICE CLASS MAY BE DELAYED WAITING FOR INITIATOR/SCHEDULER

  The APPCFEED Service Class might have failed to achieve its performance
  goal because of queue delays (either batch jobs were waiting for an
  initiator, APPC transactions were waiting for the MVS/APPC transaction
  scheduler, etc.).  The below information shows the average number of
  address spaces in the system, by category.

                          -------THIS SERVICE CLASS-------       AVERAGE
                          AVERAGE QUEUE    AVERAGE     AVG        BATCH
  MEASUREMENT INTERVAL    TIME PER JOB   JOBS QUEUED   MPL     INITIATORS
```

Suggestion: CPExpert suggests that you review your initiator structure or APPC scheduling parameters. Depending upon the amount of delay and the importance of the workload, you may wish to revise the initiator structure (provide more initiators, change the class assigned to initiators), or change the APPC scheduling.

Alternatively, you may wish to change the job class assigned to the batch workload missing the performance goal.

Rule WLM350: I/O activity may have caused significant delays

Finding: CPExpert believes that I/O activity by the service class may be a significant cause of the service class missing its performance goal.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

This finding applies only to service classes with response goals. The finding does not apply to service classes with an execution velocity goal. This is because execution velocity is computed based on CPU Using and delays for either the CPU or processor storage. I/O delays are not a part of the computation of execution velocity. Consequently, it is not possible to relate I/O delay to execution velocity.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of I/O activity and the delay to the service class caused by the I/O activity.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300:	Service Class was delayed for UNKNOWN delay
Rule WLM301:	Server Service Class was delayed for UNKNOWN delay

Discussion: When CPExpert detects that a service class did not achieve its response goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of UNKNOWN delay is that the service class was delayed because of I/O activity.

- For service classes which are assigned address spaces (that is, the service classes are **not** transactions managed by a work manager), the SRM does not collect I/O delay information. Rather, any I/O delay is reflected in the UNKNOWN category of delay.

  For these service classes, CPExpert must estimate the I/O delay based on information from SMF Type 72 records and SMF Type 74 records (and potentially, SMF Type 30 records). This rule (Rule WLM350) describes these situations.

- For service classes which describe transactions managed by a work manager (possible with CICS/ESA Version 4.1 and IMS/CICS Version

5), the work manager provides the Workload Manager with information about I/O delays from the perspective of the work manager. This situation is described in Rule WLM124.

When the UNKNOWN delay is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE), CPExpert analyzes several possible causes of delay outside the control of the SRM. Initially, CPExpert examines the I/O counts contained in the SMF Type 72 records for the measurement interval. CPExpert divides the I/O service units (R723CIOC) by the I/O service coefficient (SMF72ISD). This yields the total number of I/O operations for the service class during the measurement interval.

CPExpert cannot tell from the Type 72 information whether the I/O operations were directed to tape, to DASD, or to other device types. However, DASD normally is the fastest medium. If the I/O had been directed to DASD, the delay normally would be less than if the I/O had been directed to other activity. CPExpert makes an assumption that all I/O activity had been directed to DASD, simply to get a "feel" as to whether the I/O activity could be a significant cause for delay.

If the DASD Component of CPExpert is **not** licensed, CPExpert uses the average I/O response time in the measurement interval, for all DASD devices in the configuration.

- CPExpert processes the Type 74 records for DASD devices and computes the overall average device characteristics (I/O response, disconnect time, connect time, PEND time, and I/O Supervisor queue time) for all DASD devices.

- The overall average DASD I/O response time is multiplied by the number of I/O operations generated by the service class missing its performance goal. The result is an estimate of the maximum I/O delay using the overall average DASD response time.

If the DASD Component of CPExpert is licensed and if the CPExpert modification has been made to MXG or MICS to collect Type 30(DD) information for service classes, CPExpert can focus on specific DASD devices used by the service class missing its performance goal.

- CPExpert processes the DASD30DD records created by the modification to MXG or MICS, extracting DASD device information for the service class missing its performance goal.

- CPExpert then processes the Type 74 records for the DASD devices referenced by the service class. CPExpert extracts the device characteristics (I/O response, disconnect time, connect time, PEND

time, and I/O Supervisor queue time) for each DASD device referenced by the service class.

- The DASD I/O response time for each device referenced by the service class is multiplied by the number of I/O operations directed to the DASD device to yield an estimate of the I/O delay for each device. CPExpert sums the estimated delays for each device referenced by the service class to yield an overall estimated maximum DASD delay.

CPExpert produces Rule WLM350 if the estimated maximum DASD delay is greater than the actual response time multiplied by the **WLMSIG** guidance variable. CPExpert provides information showing the average I/O operations per transaction (from the Type 72 records for the service class), the estimated total maximum DASD delay time, and the DASD I/O characteristics during the measurement interval (I/O response, disconnect time, connect time, PEND time, and I/O Supervisor queue time).

There are several considerations with this analysis approach:

- The I/O operations counted in the Type 72 records may not have been directed to DASD. If the I/O operations were directed to some other medium (e.g., the were directed to tape), the analysis might significantly underestimate the effect of I/O on performance. This is because tape I/O operations often are much longer than DASD I/O operations. Consequently, CPExpert might not be produced Rule WLM350 if the estimated DASD I/O time were less than the significance factor. Unfortunately, there is no information at present which describes tape I/O delays[1].

- The DASD I/O operations might be buffered or overlapped with each other. This is quite likely to be the case if the service class handles batch jobs, for example. This situation is less likely with TSO interactive transactions, as TSO interactive transactions often execute few I/O operations and these are often unbuffered and unoverlapped.

- If the overall average DASD I/O response time is used, it may be that the service class referenced DASD devices which experienced I/O response times significantly different from the overall average. This situation would not occur if the CPExpert modification had been made to MXG or MICS, as CPExpert would use DASD I/O information only for the devices referenced by the service class.

[1]We are considering changing the DASD Component modification to MXG or MICS to collect tape I/O information. If this change is implemented, tape I/O information could be available.

- Even if the CPExpert modification has been made to MXG or MICS to collect DASD I/O activity by device, the Type 30 I/O activity counts may not relate well to actual DASD activity due to inconsistencies in how the Type 30 I/O counts are provided to SMF by subsystems.

As a result of the above considerations, the results of the DASD I/O analysis must be viewed with some caution. However, analysts are mostly interested in finding **significant** delays.

- If Rule WLM350 shows that the estimated I/O delays are **very significant**, it is quite likely that I/O delays are indeed accounting for much of the UNKNOWN delay. The I/O delay may not be caused by DASD but could be caused by some other (slower) medium.

- If Rule WLM350 is **not** be produced for a service class with a response goal, you can be resonably confident that DASD I/O operations are not significantly delaying the service class. If tape (or other relatively slow medium) is causing I/O delays, the service class likely describes batch jobs or long-running started tasks. These service classes do not normally have response goals and thus would not be analyzed in Rule WLM350 code.

- The "bottom line" is that when Rule WLM350 is produced, it is pretty likely that DASD I/O is significantly delaying the response time of the associated service class. The actual data reported may be suspect, but the overall finding likely is correct.

The following example illustrates the output from Rule WLM350:

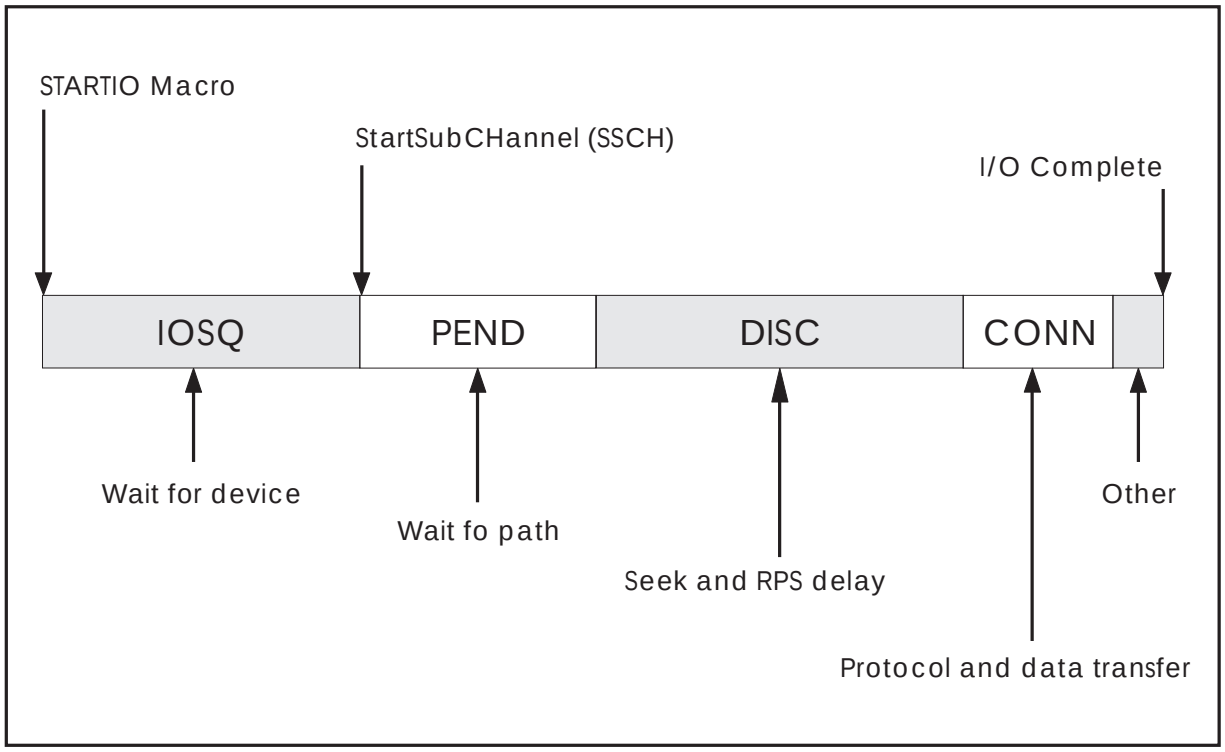
```
RULE WLM350: I/O ACTIVITY MAY HAVE CAUSED SIGNIFICANT DELAYS

   A significant part of the UNKNOWN delay probably can be attributed to
   I/O delay.  CPExpert used the average DASD I/O response time during the
   times when Service Class TSO (Period 1) missed its service goal. The
   average DASD I/O response time was multiplied by the average number of
   I/O operations per transaction to estimate the potential delay which
   might be caused by I/O activity.  The below data shows intervals when
   DASD I/O delay could have caused TSO to miss its service goal:

                          AVERAGE   ESTIMATED
                         I/O COUNT TOTAL DASD  ---AVERAGE DASD I/O TIMES--
   MEASUREMENT INTERVAL  PER TRANS TIME/TRANS  RESP  DISC  CONN  PEND  IOSQ
   13:02-13:07,21JUN1994       5      0.028   0.006 0.003 0.002 0.000 0.000
   13:07-13:12,21JUN1994       5      0.044   0.009 0.004 0.003 0.000 0.002
   13:17-13:22,21JUN1994       5      0.045   0.010 0.004 0.005 0.000 0.000
   13:22-13:27,21JUN1994       5      0.047   0.009 0.004 0.005 0.000 0.000
```

Suggestion: From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. These measures are reported by RMF in SMF Type 74 records.

The following figure illustrates these four measures and another potential element of DASD I/O time, titled "Other".



- **IOSQ time**. IOSQ time is the time from the issuance of a STARTIO macro until the Start SubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy. If the device is not busy with this system, the SSCH instruction is issued. However, if the device is busy with this system, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from this system. (The emphasis on "this system" is explained in the below discussion of PEND time.)

  Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by high seeking, high RPS delays, or high PEND time). Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

- **PEND time**. PEND time is the time from the issuance of the SSCH instruction until the device is selected by the control unit. This time is caused by queuing for the path (wait for channel, wait for control unit or wait for head-of-string), and can be caused by other systems sharing the device (wait for device). Large PEND times usually involve the following situations:

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets should be moved to a non-shared device.

    If the data sets are required to be shared, then they must be relocated to shared devices.

  - Multiple applications or users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by

contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement.

- Multiple application systems may be using the volume experiencing high PEND times. In this case, perhaps application redesign or scheduling can solve the problem.

Additionally, large PEND times for shared devices could be caused by RESERVE from the other system. The applications issuing the RESERVE should be examined to determine whether the RESERVE is required. If the RESERVE **is** required, the above situations should reviewed to determine whether improvements can be achieved.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or head-of-string. The data sets can be redistributed among different volumes on different paths, control units, or heads-of-string. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) head-of-string or path.

  If redistributing the data sets or moving the volume is not feasible, then the device should have more paths. Depending upon the existing configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices. Fairchild [2] lists four ways in which staging in caching controllers can cause hidden device busy (with the device busy potentially reflected in high PEND time):

  - The normal (random) caching algorithm stages all records to the end of the track after a requested record is read.

  - The normal (random) caching algorithm stages all records from the beginning of the track to the requested record if a front-end miss occurs.

[2] Fairchild, Bill, "The Anatomy of an I/O Request", *Conference Proceedings*, CMG'90, the Computer Measurement Group, Chicago, IL.

- The sequential caching algorithm stages all records to the end of the track after the requested record is read, and stages in all of the next track. IBM-3990 (Model 3) controllers stages in all of the next three tracks.

- Most writes to extended function IBM-3990 (Model 3) go into NVS with a subsequent destaging required.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. In this case, the dual copy initialize should be turned off.

- **DISC time**. DISC time is the time (1) from when the controller initiates a SEEK Channel Command Word (and the seek requires an arm movement) on the device until the SEEK command is complete, (2) plus the time of the rotational delay while the SET SECTOR Channel Command Word is executing, and (3) plus the rotational position sensing (RPS) delay time required because of missed RPS reconnect.

- **Seek delay**. The SEEK command is responsible for positioning the arm to the proper cylinder. If no positioning is required (that is, the arm is already at the proper cylinder), the device is not disconnected. Seek operations occur because of accessing patterns with data sets and because of accessing patterns between data sets. Large seek times usually involve the following situations:

  - Multiple data sets being active on the volume. The data sets can be redistributed among different volumes, to eliminate the seeking on the single volume.

  - Multiple users using the same data set on the volume. While only one data set is involved, the user or application accessing patterns may require frequent arm movement. A partitioned data set in which several TSO users reference different members is a common situation.

    Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the seeking problems.

- **Rotational delay**. The SET SECTOR command is responsible for locating the proper sector on the track as the disk rotates. (Actually, the SET SECTOR command locates a sector three sectors preceding the desired sector. This sector is called the *angular sector*.)

The device is disconnected during the SET SECTOR command operation.

The rotational delay may be from zero, to the total time required to rotate the disk to the required sector. It is possible that the required sector will be immediately under the head. In this case there is zero rotational delay. On the other hand, the sector could have just passed under the head before the SET SECTOR command was received by the drive. In this case, a full rotation must be accomplished before the required sector is located. On average, one-half of the rotation time will be required to locate the sector. This time is referred to as the **average latency** of the device. For example, IBM-3380 devices rotate every 16.6 milliseconds and the average latency is 8.3 milliseconds.

It is important to realize that the latency is an average based upon many SET SECTOR commands. Any particular SET SECTOR command may have a latency ranging from zero to the maximum rotational delay.

If there are few I/O commands for a particular device in a given measurement interval, it is uncertain what the average latency will be. However, if there are many I/O commands for a particular device in a given measurement interval, the average latency will normally be one-half of the rotational delay.

The average latency may be (and should be) quite small with cached devices. This is because many I/O requests should be satisfied from the cache and have no latency.

- **Missed RPS reconnect**. The device attempts to reconnect to the path when the angular sector is reached (the angular sector is described above). If the reconnect attempt is successful before the desired sector is reached, then the device connects and the read or write operation can proceed. If the reconnect is not successful before the desired sector is reached, then the device does not connect, and a complete revolution of the track must occur before the angular sector is again reached. This is called a *missed rotational position sensing reconnect (or missed RPS reconnect)* delay.

  There is no action which can alleviate the initial rotational positioning delay (aside from changing device characteristics, such as implementing caching or buffering at the device level). Over a large number of I/O operations, this initial delay will be one-half the rotation times.

However, the missed RPS delay is a function of the probability that the path will be busy when the device attempts to reconnect; the busier the path(s), the more missed RPS delay. (Note that the path busy time is a function of the connect time of other actuators. The path cannot be busy from the device itself when the device attempts to reconnect.)

- **CONN time**. CONN time is the time in which the device is actually connected to the path. This time includes the data transfer time, but also includes protocol exchange[3] (or "hand shaking") between the various components at several stages of the I/O operation.

The data transfer time obviously is a function of the amount of data being transferred. This simply is the number of bytes transferred divided by the transfer speed (for example, if 4096 bytes were transferred from an IBM-3390 with a transfer speed of 4,200,000 bytes per second, the 4096 bytes would require 4096/4,200,000 seconds; or about 0.975 milliseconds).

Large connect times generally are caused by the following situations:

- A large average block size. This situation may be highly desireable for sequential data sets, but would be quite undesirable for randomly accessed data.

- Long multi-track searches. For example, the catalog must be searched for cataloged files, the Volume Table of Contents (VTOC) must searched to find a requested file, a directory must be searched for partitioned data sets, etc.. These searches will result in long connect times for the volume involved.

- Fairchild indicates that wrong sector numbers, non-IBM temporary error recovery, and coding OPTCD=W can also cause long connect times.

There have been no published "official" times for the protocol connect times. The protocol time has been estimated by different authors to range from .5 milliseconds to 1.5 milliseconds, depending upon the device and the authors' experiences. In any event, there generally is little that can be done about the protocol time.

- **OTHER time**. There are at least two other potential I/O delays for DASD: (1) waiting for the I/O completion interrupt to be serviced by

[3]Note that the protocol exchange occurs at multiple points in the normal I/O operation, even though it is shown only once in Exhibit 5-1.

a processor and (2) waiting for the I/O interrupt to be serviced by a domain under PR/SM. Neither potential I/O delay is expected to be of the magnitude of the four "standard" I/O delays. However, they can be significant in special circumstances.

- Multi-processor configurations running under MVS/XA and MVS/ESA can use any processor to service an I/O interrupt. However, when a processor services an I/O interrupt, the processor's high-speed cache storage is no longer valid when control is returned to the interrupted task. Consequently, many of the processor's high-performance design features may be nullified.

  A hardware feature allows processors to be disabled for I/O interrupts. With this method, only a small number (perhaps only one) processor is enabled for interrupt processing. Only this processor will have its high-speed cache storage disturbed by the task-switching required for interrupt processing, and only this processor will periodically have its high-performance design features nullified. The disadvantage to this approach is that an interrupt may occur while the processor is busy servicing a previous interrupt.

  If an interrupt is pending and no processor is enabled to service the interrupt, the interrupt must wait until a processor is available. This time should be insignificant, unless the system is processing a significantly large number of I/O operations. If the system is processing a large number of I/O operations, the interrupt pending delay could pose performance problems. After the processor completes processing for an I/O interrupt, it issues a Test Pending Interrupt (TPI) instruction to determine whether there are any interrupts pending. If an I/O interrupt is pending, the processor proceeds to service that interrupt.

  The CPENABLE keyword in the IEAOPTxx member of SYS1.PARMLIB is used to specify the percent of I/O interrupts detected by the TPI instruction, compared with all I/O interrupts. When the percent exceeds the high threshold of the CPENABLE keyword, MVS enables another processor to handle pending I/O interrupts. If the percent falls below the low threshold of the CPENABLE keyword, MVS will disable a processor (to the point that only one processor is enabled).

- MVS environments running under as a guest under VM or in a logical partition (LPAR) under PR/SM are subject to I/O interrupt delays. These delays can occur if another guest (for VM) or another domain is in its dispatch interval when the I/O interrupt completion is posted. The I/O interrupt remains pending until the guest or domain is

dispatched. These delays have been estimated to be far more significant than might otherwise be expected.

Rule WLM351: I/O activity may have caused significant delays

Finding: CPExpert believes that I/O activity by the service class may be a significant cause of the service class missing its performance goal. This finding is produced when (1) the DASD Component of CPExpert is licensed and (2) the CPExpert modification to MXG or MICS has been installed to associate device information to service classes.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of I/O activity and the delay to the service class caused by the I/O activity.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM300:	Service Class was delayed for UNKNOWN delay
Rule WLM301:	Server Service Class was delayed for UNKNOWN delay

Discussion: Rule WLM351 is similar to Rule WLM350, except that more precise information is presented. Rule WLM351 is invoked if the DASD Component of CPExpert is licensed and the CPExpert modification to MXG or MICS has been installed to associate device information to service classes.

The CPExpert modification to MXG or MICS records the system, job, job step, service class, and summary information about each job step's use of DASD devices by device number. The information is recorded only at the device level (rather than at the DD name level) and only required information is retained. Consequently, the records are small and use only a small amount of DASD space.

The WLM Component of CPExpert can use the information described above to identify the **specific** devices referenced by any service class which misses its performance goal. The I/O characteristics of these devices are used by CPExpert to assess the likely impact of I/O delays on the response time of the service class.

©Copyright 1994, Computer Management Sciences, Inc.

Suggestion: Please refer to Rule WLM350 for a discussion of I/O delays and alternatives to reduce the delays.

Rule WLM352: I/O activity may have caused significant delays to server

Finding: CPExpert believes that I/O activity by the service class may be a significant cause of the service class missing its performance goal. This finding is produced when (1) the DASD Component of CPExpert is licensed and (2) the CPExpert modification to MXG or MICS has been installed to associate device information to service classes, but SMF Type 30(Interval) records **were not** available during the measurement intervals when the service class missed its performance goal.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of I/O activity and the delay to the service class caused by the I/O activity.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM301: Server Service Class was delayed for UNKNOWN delay

Discussion: When CPExpert produces Rule WLM301 to indicate that a server service class was delayed for UNKNOWN reasons, CPExpert attempts to estimate the amount of UNKNOWN delay which might be attributed to I/O delay.

Rule WLM352 is similar to Rule WLM350, except the information pertains to server service classes.

Suggestion: Please refer to Rule WLM350 for a discussion of I/O delays and alternatives to reduce the delays.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM353: I/O activity may have caused significant delays to server

Finding: CPExpert believes that I/O activity by the service class may be a significant cause of the service class missing its performance goal. This finding is produced when (1) the DASD Component of CPExpert is licensed and (2) the CPExpert modification to MXG or MICS has been installed to associate device information to service classes, and (3) SMF Type 30(Interval) records for the address spaces in the server service class **were** available during the measurement intervals when the service class missed its performance goal.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of I/O activity and the delay to the service class caused by the I/O activity.

Logic flow: The following rule causes this rule to be invoked:

Rule WLM301: Server Service Class was delayed for UNKNOWN delay

Discussion: When CPExpert produces Rule WLM301 to indicate that a server service class was delayed for UNKNOWN reasons, CPExpert attempts to estimate the amount of UNKNOWN delay which might be attributed to I/O delay.

Rule WLM353 is similar to Rule WLM350, except the information pertains to server service classes.

Suggestion: Please refer to Rule WLM350 for a discussion of I/O delays and alternatives to reduce the delays.

©Copyright 1994, Computer Management Sciences, Inc.

Rule WLM355: Device DISC time was a major cause of DASD I/O delay

Finding: CPExpert has determined that device DISC time was a major cause of delay in DASD response for the I/O operations of the service class.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM350:	I/O activity may have caused significant delays
Rule WLM351:	I/O activity may have caused significant delays
Rule WLM352:	I/O activity may have caused significant delays for server service class
Rule WLM353:	I/O activity may have caused significant delays for server service class

Discussion: DISC time is the time (1) from when the controller initiates a SEEK Channel Command Word (and the seek requires an arm movement) on the device until the SEEK command is complete, (2) plus the time of the rotational delay while the SET SECTOR Channel Command Word is executing, and (3) plus the rotational position sensing (RPS) delay time required because of missed RPS reconnect. Please see Rule WLM350 for a discussion of DISC time.

CPExpert computes the average per-second DISC delay time as described in Rule WLM350. Rule WLM355 is produced if the average DISC time accounted for a significant percent of the response time of transactions in the service class missing its performance goal.

The following example illustrates the output from Rule WLM355:

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM355: DEVICE DISCONNECT TIME WAS A MAJOR CAUSE OF DASD DELAYS

   A major part of the potential I/O delay to the TSO Service Class
   could be attributed to device disconnect (DISC) time.  Disconnect time
   is caused by seeking or missed rotational position reconnect (RPS).
   Please refer to the WLM Component User Manual for advice on how to
   minimize device DISC time.
```

Suggestion: Large DISC time is caused by device seek delay, device rotational delay, or device missed RPS reconnect delay.

- **Seek delay**. The SEEK command is responsible for positioning the arm to the proper cylinder. If no positioning is required (that is, the arm is already at the proper cylinder), the device is not disconnected. Seek operations occur because of accessing patterns with data sets and because of accessing patterns between data sets. Large seek times usually involve the following situations:

  - Multiple data sets being active on the volume. The data sets can be redistributed among different volumes, to eliminate the seeking on the single volume.

  - Multiple users using the same data set on the volume. While only one data set is involved, the user or application accessing patterns may require frequent arm movement. A partitioned data set in which several TSO users reference different members is a common situation.

    Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the seeking problems.

- **Rotational delay**. The SET SECTOR command is responsible for locating the proper sector on the track as the disk rotates. (Actually, the SET SECTOR command locates a sector three sectors preceding the desired sector. This sector is called the *angular sector*.) The device is disconnected during the SET SECTOR command operation.

  The rotational delay may be from zero, to the total time required to rotate the disk to the required sector. It is possible that the required sector will be immediately under the head. In this case there is zero rotational delay. On the other hand, the sector could have just passed under the head before the SET SECTOR command was received by the drive. In this case, a full rotation must be accomplished before the required sector is located. On average, one-half of the rotation time will be required to

locate the sector. This time is referred to as the **average latency** of the device. For example, IBM-3390 devices rotate every 14.1 milliseconds and the average latency is 7.1 milliseconds[1].

It is important to realize that the latency is an average based upon many SET SECTOR commands. Any particular SET SECTOR command may have a latency ranging from zero to the maximum rotational delay.

If there are few I/O commands for a particular device in a given measurement interval, it is uncertain what the average latency will be. However, if there are many I/O commands for a particular device in a given measurement interval, the average latency will normally be one-half of the rotational delay.

The average latency may be (and should be) quite small with cached devices. This is because many I/O requests should be satisfied from the cache and have no latency.

If the average DISC per I/O operation is approximately equal to the average latency for the device type, the device would benefit from caching. If the average DISC per I/O operation is much larger than the average latency for the device type, it is unclear whether the device would benefit from caching.

- **Missed RPS reconnect**. The device attempts to reconnect to the path when the angular sector is reached (the angular sector is described above). If the reconnect attempt is successful before the desired sector is reached, then the device connects and the read or write operation can proceed.

  A path must be available for the device to reconnect successfully. If the reconnect is not successful before the desired sector is reached, then the device does not connect, and a complete revolution of the track must occur before the angular sector is again reached. This is called a *missed rotational position sensing reconnect (or missed RPS reconnect)* delay.

  There is no action which can alleviate the initial rotational positioning delay (aside from changing device characteristics, such as implementing caching or buffering at the device level). Over a large number of I/O operations, this initial delay will be one-half the rotation times.

  However, the missed RPS delay is a function of the probability that the path will be busy when the device attempts to reconnect; the busier the

[1]These values apply only to Model 1, Model 2, and Model 3. IBM-3390 Model 9 devices rotate every 45.1 milliseconds and the average latency is 22.8 milliseconds.

path(s), the more missed RPS delay. (Note that the path busy time is a function of the connect time of other actuators. The path cannot be busy from the device itself when the device attempts to reconnect.).

Rule WLM356: Device PEND time was a major cause of DASD I/O delay

Finding: CPExpert has determined that device PEND time was a major cause of delay in DASD response for the I/O operations of the service class.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM350:	I/O activity may have caused significant delays
Rule WLM351:	I/O activity may have caused significant delays
Rule WLM352:	I/O activity may have caused significant delays for server service class
Rule WLM353:	I/O activity may have caused significant delays for server service class

Discussion: PEND time is the time from the issuance of the SSCH instruction until the device is selected by the control unit. This time is caused by queuing for the path (wait for channel, wait for control unit or wait for head-of-string), and can be caused by other systems sharing the device (wait for device).

CPExpert computes the average per-second PEND delay time as described in Rule WLM350. Rule WLM356 is produced if the average PEND time accounted for a significant percent of the response time of transactions in the service class missing its performance goal.

The following example illustrates the output from Rule WLM356:

```
RULE WLM356: DEVICE PEND TIME WAS A MAJOR CAUSE OF DASD DELAYS

   A major part of the potential I/O delay to the ST_USER Service Class
   could be attributed to device pending (PEND) time.  Pending time is
   caused by queuing for the path (wait for channel, wait for control unit
   or wait for head-of-string).  The queuing can be caused by other systems
   sharing the device (wait for device).  Large PEND times for devices that
   are not shared may mean that there are insufficient paths available to
   the device.  Please refer to the WLM Component User Manual for advice on
   how to minimize device PEND time.
```

©Copyright 1994, Computer Management Sciences, Inc.

Suggestion: Large device PEND times usually involve the following situations:

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - **Multiple data sets active on the volume**. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets should be moved to a non-shared device.

    If the data sets are required to be shared, then they must be relocated to shared devices.

  - **Multiple applications or users using the same data set on the volume**. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement.

  - **Multiple application systems may be using the volume experiencing high PEND times**. In this case, perhaps application redesign or scheduling can solve the problem.

  Additionally, large PEND times for shared devices could be caused by RESERVE from the other system. The applications issuing the RESERVE should be examined to determine whether the RESERVE is required. If the RESERVE **is** required, the above situations should reviewed to determine whether improvements can be achieved.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or head-of-string. The data sets can be redistributed among different volumes on different paths, control units, or heads-of-string. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) head-of-string or path.

©Copyright 1994, Computer Management Sciences, Inc.

If redistributing the data sets or moving the volume is not feasible, then the device should have more paths. Depending upon the existing configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices. Fairchild [1] lists four ways in which staging in caching controllers can cause hidden device busy (with the device busy potentially reflected in high PEND time):

  - The normal (random) caching algorithm stages all records to the end of the track after a requested record is read.

  - The normal (random) caching algorithm stages all records from the beginning of the track to the requested record if a front-end miss occurs.

  - Most writes to extended function IBM-3990 (Model 3) go into NVS with a subsequent destaging required.

  - The sequential caching algorithm stages all records to the end of the track after the requested record is read, and stages in all of the next track. IBM-3990 (Model 3) controllers stages in all of the next three tracks.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. In this case, the dual copy initialize should be turned off.

[1] Fairchild, Bill, "The Anatomy of an I/O Request", *Conference Proceedings*, CMG'90, the Computer Measurement Group, Chicago, IL.

Rule WLM357: CONNECT TIME WAS A MAJOR CAUSE OF I/O DELAY

Finding: Connect time was a major cause of the I/O delay with the volume.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding may have a LOW IMPACT or MEDIUM IMPACT on the performance of the device.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM350:	I/O activity may have caused significant delays
Rule WLM351:	I/O activity may have caused significant delays
Rule WLM352:	I/O activity may have caused significant delays for server service class
Rule WLM353:	I/O activity may have caused significant delays for server service class

Discussion: Connect time is the time in which the device is actually connected to the path. This time includes the data transfer time, but also includes protocol exchange (or "hand shaking") between the various components at several stages of the I/O operation.

The data transfer time obviously is a function of the amount of data being transferred. This simply is the number of bytes transferred divided by the transfer speed (for example, if 4096 bytes were transferred from an IBM-3380 with a transfer speed of 3,000,000 bytes per second, the 4096 bytes would require 4096/3,000,000 seconds; or about 1.36 milliseconds).

Large connect times generally are caused by the following situations:

- A large average block size. This situation may be highly desirable for sequential data sets, but would be undesirable for randomly accessed data.

- Long multi-track searches. For example, the catalog must be searched for cataloged files, the Volume Table of Contents (VTOC) must searched to find a requested file, a directory must be searched for partitioned data sets, etc.. These searches will result in long connect times for the volume involved.

©Copyright 1994, Computer Management Sciences, Inc.

- Program loading from system packs.

The following example illustrates the output from Rule WLM357:

```
RULE WLM357: DEVICE CONNECT TIME WAS A MAJOR CAUSE OF DASD DELAYS

   A major part of the potential I/O delay to the TSO Service Class
   could be attributed to device connect (CONN) time.  Connect time is
   caused primarily by data transfer.  Please refer to the WLM Component
   User Manual for advice on how to minimize device connect time.
```

Suggestion: As mentioned above, large connect times may be acceptable, depending upon the nature of the application files and why the large connect times occur.

CPExpert suggests that you review the files accessed by the service class missing its performance goal. Based on this review, you can decide whether the large connect times are appropriate or whether action should be take with respect to the application files.

- If the large connect times are appropriate, you may wish to review the performance goal specified for the service class. Depending upon how much connect time was responsible for the I/O delay (and how much the I/O delay accounted for the service class missing its performance goal), you may wish to adjust the performance goal. This, of course, is the easiest solution: you simply adjust the performance goal considering the data transfer requirements of the applications.

- If the large connect times are not appropriate (or if you cannot adjust the performance goal because of management decisions), you may be required to address the application and its files. This step may require considerable effort, depending upon the application, and normally will not be taken lightly.

Rule WLM358: QUEUING IN IOS WAS A MAJOR CAUSE OF I/O DELAY

Finding: Queuing in the I/O Supervisor (IOSQ) was a major cause of the I/O delay with the volume.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the device.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM350:	I/O activity may have caused significant delays
Rule WLM351:	I/O activity may have caused significant delays
Rule WLM352:	I/O activity may have caused significant delays for server service class
Rule WLM353:	I/O activity may have caused significant delays for server service class

Discussion: IOSQ time is the time from the issuance of a STARTIO macro until the Start SubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy. If the device is not busy with this system, the SSCH instruction is issued. However, if the device is busy with this system, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from this system.

Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by high seeking, high RPS delays, or high PEND time).

The following example illustrates the output from Rule WLM358:

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM358: DEVICE IOS QUEUING TIME WAS A MAJOR CAUSE OF DASD DELAYS

  A major part of the potential I/O delay to the TSO Service Class
  could be attributed to queuing in the I/O Supervisor (IOSQ).  IOSQ
  time is caused by too many I/O operations directed to the device or
  lengthy device response times (perhaps caused by high seeking, by high
  RPS delays, or by high PEND time.  Please refer to the WLM Component
  User Manual for advice on how to minimize device IOSQ time.
```

Suggestion: Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

Depending on the amount of IOSQ time involved, on budget considerations, and on the business importance of the work being delayed, you might consider acquiring Parallel Access Volumes (PAV). The PAV design tends to eliminate IOSQ time.

Rule WLM359: I/O ACTIVITY PROBABLY DID NOT CAUSE MAJOR DELAYS

Finding: I/O activity probably was not a significant factor in the UNKNOWN delay.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding has NO IMPACT. The finding is produced for information purposes.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300:	Service Class was delayed for UNKNOWN delay
Rule WLM301:	Server Service Class was delayed for UNKNOWN delay

Discussion: As described in the above rules, the UNKNOWN category of workload delay means that the Workload Manager was unable to identity the cause of the delay. The delay normally is caused by something over which the System Resources Manager has no control. This delay category potentially includes I/O delay, ENQ delay, etc.

CPExpert estimates the amount of the delay which might have been attributed to I/O operations. The process by which CPExpert makes the estimate is described in Rule WLM350. CPExpert produces Rule WLM350 if the I/O activity might have caused significant delays.

CPExpert produces Rule WLM359 if the I/O activity probably did not cause significant delays. The purpose of Rule WLM359 is to alert you to the possibility of other factors which may cause the UNKNOWN delay.

The following example illustrates the output from Rule WLM359:

©Copyright 1994, Computer Management Sciences, Inc.

```
RULE WLM359: I/O ACTIVITY PROBABLY DID NOT CAUSE MAJOR DELAYS

   DASD activity probably did not account for much of the UNKNOWN delay
   when Service Class TPNSODD (Period 6) missed its service goal.  The
   average DASD I/O response time was multiplied by the average number
   of I/O operations per transaction to estimate the potential delay which
   might be caused by I/O activity.  The below data shows intervals when
   DASD I/O delay apparently was not a significant in causing TPNSODD to
   miss its service goal:

                         AVERAGE   ESTIMATED
                        I/O COUNT TOTAL DASD  ---AVERAGE DASD I/O TIMES---
   MEASUREMENT INTERVAL  PER TRANS TIME/TRANS  RESP  DISC  CONN  PEND  IOSQ
   15:00-15:16,01MAR1994       0       0.002  0.006 0.002 0.003 0.001 0.000
```

Note that the "AVERAGE I/O COUNT PER TRANS" is shown as "0" in the example, while the "AVERAGE DASD I/O TIMES" columns show values. This is because of the precision of the printed results. The average I/O count per transaction actually was a very small value and rounding produced "0" as the value (that is, less than half of the transactions issued an I/O instruction recorded by SMF).

Suggestion: This finding is produced simply for information purposes.

Rule WLM360: SERVICE CLASS DID NOT REFERENCE DASD

Finding: The service class which missed its performance goal was delayed for an UNKNOWN delay. In many situations, this delay will be caused by I/O operations. However, the service class did not reference DASD and CPExpert can state that DASD delay was not a part of the UNKNOWN delay. This finding is produced only if the CPExpert modification has been made to MXG or MICS to collect DASD information for service classes.

This finding applies only to MVS versions prior to OS/390 Release 3, and to MVS versions with OS/390 Release 3 if I/O Priority Management has **not** been specified.

Impact: This finding has NO IMPACT. The finding is produced for information purposes

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300:	Service Class was delayed for UNKNOWN delay
Rule WLM301:	Server Service Class was delayed for UNKNOWN delay

Discussion: As described in the above rules, the UNKNOWN category of workload delay means that the Workload Manager was unable to identity the cause of the delay. The delay normally is caused by something over which the System Resources Manager has no control. This delay category potentially includes I/O delay, ENQ delay, etc.

CPExpert estimates the amount of the delay which might have been attributed to I/O operations. The process by which CPExpert makes the estimate is described in Rule WLM350. CPExpert produces Rule WLM350 if the I/O activity might have caused significant delays.

If the DASD Component of CPExpert is licensed and if the CPExpert modification has been made to MXG or MICS to collect Type 30(DD) information for service classes, CPExpert can focus on only the DASD devices used by the service class missing its performance goal.

CPExpert processes the DASD30DD records created by the modification to MXG or MICS, extracting DASD device information for the service class missing its performance goal. **Please note that the Type 30 (Interval) records do not include VSAM I/O references**.

©Copyright 1994, Computer Management Sciences, Inc.

CPExpert detects situations in which the service class did not reference DASD. The purpose of Rule WLM360 (this rule) is to advise you that the service class did not reference DASD using normal I/O operations. The service class might have referenced VSAM files (which, of course, reside on DASD). However, the SMF Type 30 (Interval) records do not include VSAM I/O references. Thus, CPExpert has no information on the VSAM references[1] by the service class missing its performance goal.

CPExpert produces Rule WLM360 (this rule) if the DASD I/O activity probably did not cause significant delays. The purpose of Rule WLM360 is to alert you to the possibility of other factors which may cause the UNKNOWN delay.

The following example illustrates the output from Rule WLM360:

```
RULE WLM360: SERVICE CLASS DID NOT REFERENCE DASD

  Service Class TPNSEVEN (Period 1) apparently did not reference DASD
  during the below measurement intervals, as no DASD information was
  in the SMF Type 30(DD) records collected by the CPExpert modification
  to MXG.  The SMF Type 72 records did reflect I/O activity, but the
  I/O activity was to non-DASD devices.  This I/O activity could have
  caused TPNSEVEN to miss its response goal, but CPExpert does not have
  sufficient information on which to base such a conclusion.

                                 SMF TYPE 72      TOTAL    AVG  EXCP
   MEASUREMENT INTERVAL        TOTAL EXCP COUNT   TRANS    PER TRANS
   13:02-13:07,21JUN1994           18,144           672        27
```

Rule WLM360 shows the total I/O activity reflected in the SMF Type 72 records. If this activity is relatively high (as is shown by the example), the service class referenced some device type other than DASD (for example, the service class referenced tape drives). At present, CPExpert does not continue analysis of the configuration.

Suggestion: This finding is produced simply for information purposes.

[1]The SMF Type 64 records do contain information about VSAM activity based on job name and VSAM catalog. The Type 64 records do not relate the information to service class. CPExpert does not analyze the Type 64 records at present, although future code may analyze this information.

Rule WLM361: Non-paging DASD I/O activity caused significant delays

Finding: Non-paging DASD I/O activity by the service class was a significant cause of the service class missing its performance goal.

This finding applies only to service classes with execution velocity goals, and then applies only if non-paging DASD I/O using and I/O delay are included in the execution velocity calculation. Rule WLM362 applies to service classes with response goals.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of non-paging DASD I/O activity and the delay to the service class caused by the non-paging DASD I/O activity.

Logic flow: The following rules cause this rule to be invoked:
Rule WLM103: Service Class did not achieve execution velocity goal

Discussion: When CPExpert detects that a service class did not achieve its execution velocity goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of delay is that the service class was delayed because of non-paging DASD I/O activity.

Non-paging DASD I/O delays are a part of the computation of execution velocity beginning with OS/390 Release 3, and only if the Workload Manager has been instructed to include I/O using and I/O delay in the calculation of execution velocity.

The SRM collects I/O using and delay information beginning with OS/390 Release 3. Prior to OS/390 Release 3, any I/O delay is reflected in the UNKNOWN category of delay, and CPExpert will analyze the I/O delay as discussed in Rule WLM350.

The non-paging DASD I/O using and delay information is reported in SMF Type 72 records for each service class period. CPExpert analyzes the non-paging DASD I/O delay (field R723CIOD) for service classes missing their performance goal. CPExpert produces Rule WLM361 when the percent delay caused by non-paging DASD I/O is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE), and an execution velocity goal has been specified..

After producing Rule WLM361, CPExpert analyzes several possible causes of non-paging DASD I/O delay and reports the result in subsequent rules.

©Copyright 1997, Computer Management Sciences, Inc.

The following example illustrates the output from Rule WLM361:

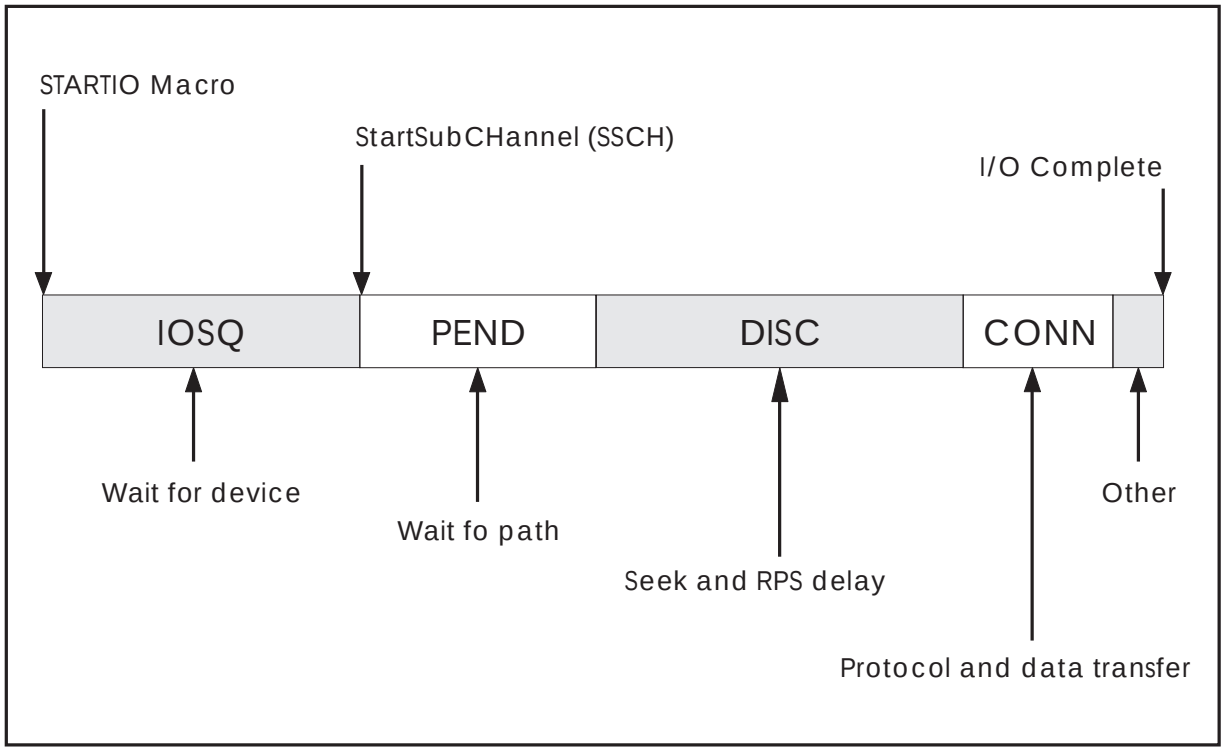
```
RULE WLM361: NON-PAGING DASD I/O ACTIVITY CAUSED SIGNIFICANT DELAYS

   IMS (Period 1): A significant part of the delay to the service
   class can be attributed to non-paging DASD I/O delay.  The below data
   shows intervals when non-paging DASD delay caused IMS to miss its
   performance goal:

                           AVG DASD   AVG DASD     --AVERAGE DASD I/O TIMES-
   MEASUREMENT INTERVAL  I/O RATE   USING/SEC    RESP   WAIT   DISC   CONN
   11:00-11:15,08MAR1997     56       0.527     0.009  0.003  0.005   0.002
```

Suggestion: From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. The last three of these measures are reported in SMF Type 72 records (fields R723CIWT, R723CIDT, and R723CICT, respectively) for environments prior to OS/390 V2R4. IOSQ time is reported in SMF Type 72 (field R723CIOT) beginning with OS/390 V2R4.

The following figure illustrates these four measures and another potential element of DASD I/O time, titled "Other".



- **IOSQ time**. IOSQ time is the time from the issuance of a STARTIO macro until the Start SubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the

device is busy. If the device is not busy with this system, the SSCH instruction is issued. However, if the device is busy with this system, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from this system. (The emphasis on "this system" is explained in the below discussion of PEND time.)

Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by high seeking, high RPS delays, or high PEND time). Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

IOSQ time is not reported in the Type 72 records prior to OS/390 V2R4, and thus is not analyzed by CPExpert for these versions of MVS. IOSQ time is reported beginning with OS/390 V2R4 and is analyzed by CPExpert beginning with this release.

- **PEND time**. PEND time is the time from the issuance of the SSCH instruction until the device is selected by the control unit. This time is caused by queuing for the path (wait for channel, wait for control unit or wait for head-of-string), and can be caused by other systems sharing the device (wait for device). Large PEND times usually involve the following situations:

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets should be moved to a non-shared device.

    If the data sets are required to be shared, then they must be relocated to shared devices.

  - Multiple applications or users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement.

  - Multiple application systems may be using the volume experiencing high PEND times. In this case, perhaps application redesign or scheduling can solve the problem.

  Additionally, large PEND times for shared devices could be caused by RESERVE from the other system. The applications issuing the RESERVE should be examined to determine whether the RESERVE is required. If the RESERVE **is** required, the above situations should reviewed to determine whether improvements can be achieved.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or head-of-string. The data sets can be redistributed among different volumes on different paths, control units, or heads-of-string. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) head-of-string or path.

If redistributing the data sets or moving the volume is not feasible, then the device should have more paths. Depending upon the existing configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices. Fairchild [1] lists four ways in which staging in caching controllers can cause hidden device busy (with the device busy potentially reflected in high PEND time):

  - The normal (random) caching algorithm stages all records to the end of the track after a requested record is read.

  - The normal (random) caching algorithm stages all records from the beginning of the track to the requested record if a front-end miss occurs.

  - The sequential caching algorithm stages all records to the end of the track after the requested record is read, and stages in all of the next track. IBM-3990 (Model 3) controllers stages in all of the next three tracks.

  - Most writes to extended function IBM-3990 (Model 3) go into NVS with a subsequent destaging required.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. In this case, the dual copy initialize should be turned off.

PEND time is accumulated along with, queue waiting for a controller, in SMF Type 72 records (field R723CIWT).

- **DISC time**. DISC time is the time (1) from when the controller initiates a SEEK Channel Command Word (and the seek requires an arm movement) on the device until the SEEK command is complete, (2) plus the time of the rotational delay while the SET SECTOR Channel Command Word is executing, and (3) plus the rotational position sensing (RPS) delay time required because of missed RPS reconnect.

[1] Fairchild, Bill, "The Anatomy of an I/O Request", *Conference Proceedings*, CMG'90, the Computer Measurement Group, Chicago, IL.

- **Seek delay**. The SEEK command is responsible for positioning the arm to the proper cylinder. If no positioning is required (that is, the arm is already at the proper cylinder), the device is not disconnected. Seek operations occur because of accessing patterns with data sets and because of accessing patterns between data sets. Large seek times usually involve the following situations:

  - Multiple data sets being active on the volume. The data sets can be redistributed among different volumes, to eliminate the seeking on the single volume.

  - Multiple users using the same data set on the volume. While only one data set is involved, the user or application accessing patterns may require frequent arm movement. A partitioned data set in which several TSO users reference different members is a common situation.

    Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the seeking problems.

- **Rotational delay**. The SET SECTOR command is responsible for locating the proper sector on the track as the disk rotates. (Actually, the SET SECTOR command locates a sector three sectors preceding the desired sector. This sector is called the *angular sector*.) The device is disconnected during the SET SECTOR command operation.

  The rotational delay may be from zero, to the total time required to rotate the disk to the required sector. It is possible that the required sector will be immediately under the head. In this case there is zero rotational delay.

  On the other hand, the sector could have just passed under the head before the SET SECTOR command was received by the drive. In this case, a full rotation must be accomplished before the required sector is located. On average, one-half of the rotation time will be required to locate the sector. This time is referred to as the **average latency** of the device. For example, IBM-3380 devices rotate every 16.6 milliseconds and the average latency is 8.3 milliseconds.

  It is important to realize that the latency is an average based upon many SET SECTOR commands. Any particular SET SECTOR command may have a latency ranging from zero to the maximum rotational delay.

If there are few I/O commands for a particular device in a given measurement interval, it is uncertain what the average latency will be. However, if there are many I/O commands for a particular device in a given measurement interval, the average latency will normally be one-half of the rotational delay.

The average latency may be (and should be) quite small with cached devices. This is because many I/O requests should be satisfied from the cache and have no latency.

- **Missed RPS reconnect**. The device attempts to reconnect to the path when the angular sector is reached (the angular sector is described above). If the reconnect attempt is successful before the desired sector is reached, then the device connects and the read or write operation can proceed.

  If the reconnect is not successful before the desired sector is reached, then the device does not connect, and a complete revolution of the track must occur before the angular sector is again reached. This is called a *missed rotational position sensing reconnect (or missed RPS reconnect)* delay.

  There is no action which can alleviate the initial rotational positioning delay (aside from changing device characteristics, such as implementing caching or buffering at the device level). Over a large number of I/O operations, this initial delay will be one-half the rotation times.

  However, the missed RPS delay is a function of the probability that the path will be busy when the device attempts to reconnect; the busier the path(s), the more missed RPS delay. (Note that the path busy time is a function of the connect time of other actuators. The path cannot be busy from the device itself when the device attempts to reconnect.)

- **CONN time**. CONN time is the time in which the device is actually connected to the path. This time includes the data transfer time, but also includes protocol exchange[2] (or "hand shaking") between the various components at several stages of the I/O operation.

The data transfer time obviously is a function of the amount of data being transferred. This simply is the number of bytes transferred divided by the transfer speed (for example, if 4096 bytes were transferred from an IBM-

[2]Note that the protocol exchange occurs at multiple points in the normal I/O operation, even though it is shown only once in Exhibit 5-1.

3390 with a transfer speed of 4,200,000 bytes per second, the 4096 bytes would require 4096/4,200,000 seconds; or about 0.975 milliseconds).

Large connect times generally are caused by the following situations:

- A large average block size. This situation may be highly desirable for sequential data sets, but would be quite undesirable for randomly accessed data.

- Long multi-track searches. For example, the catalog must be searched for cataloged files, the Volume Table of Contents (VTOC) must searched to find a requested file, a directory must be searched for partitioned data sets, etc.. These searches will result in long connect times for the volume involved.

- Fairchild indicates that wrong sector numbers, non-IBM temporary error recovery, and coding OPTCD=W can also cause long connect times.

There have been no published "official" times for the protocol connect times. The protocol time has been estimated by different authors to range from .5 milliseconds to 1.5 milliseconds, depending upon the device and the authors' experiences. In any event, there generally is little that can be done about the protocol time.

- **OTHER time**. There are at least two other potential I/O delays for DASD: (1) waiting for the I/O completion interrupt to be serviced by a processor and (2) waiting for the I/O interrupt to be serviced by a domain under PR/SM. Neither potential I/O delay is expected to be of the magnitude of the four "standard" I/O delays. However, they can be significant in special circumstances.

- Multi-processor configurations running under MVS/XA and MVS/ESA can use any processor to service an I/O interrupt. However, when a processor services an I/O interrupt, the processor's high-speed cache storage is no longer valid when control is returned to the interrupted task. Consequently, many of the processor's high-performance design features may be nullified.

  A hardware feature allows processors to be disabled for I/O interrupts. With this method, only a small number (perhaps only one) processor is enabled for interrupt processing. Only this processor will have its high-speed cache storage disturbed by the task-switching required for interrupt processing, and only this processor will periodically have its high-performance design features nullified. The

disadvantage to this approach is that an interrupt may occur while the processor is busy servicing a previous interrupt.

If an interrupt is pending and no processor is enabled to service the interrupt, the interrupt must wait until a processor is available. This time should be insignificant, unless the system is processing a significantly large number of I/O operations. If the system is processing a large number of I/O operations, the interrupt pending delay could pose performance problems.

After the processor completes processing for an I/O interrupt, it issues a Test Pending Interrupt (TPI) instruction to determine whether there are any interrupts pending. If an I/O interrupt is pending, the processor proceeds to service that interrupt.

The CPENABLE keyword in the IEAOPTxx member of SYS1.PARMLIB is used to specify the percent of I/O interrupts detected by the TPI instruction, compared with all I/O interrupts. When the percent exceeds the high threshold of the CPENABLE keyword, MVS enables another processor to handle pending I/O interrupts. If the percent falls below the low threshold of the CPENABLE keyword, MVS will disable a processor (to the point that only one processor is enabled).

- MVS environments running under as a guest under VM or in a logical partition (LPAR) under PR/SM are subject to I/O interrupt delays. These delays can occur if another guest (for VM) or another domain is in its dispatch interval when the I/O interrupt completion is posted. The I/O interrupt remains pending until the guest or domain is dispatched. These delays have been estimated to be far more significant than might otherwise be expected.

Rule WLM362: Non-paging DASD I/O activity caused significant response delays

Finding: Non-paging DASD I/O activity by the service class was a significant cause of the service class missing its response performance goal.

This finding applies only to service classes with response goals. I/O delays are a part of the computation of execution velocity only with OS/390 Release 3. Rule WLM361 applies to service classes with execution velocity goals.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of DASD I/O activity and the delay to the service class caused by the DASD I/O activity.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal

Discussion: When CPExpert detects that a service class did not achieve its response goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of delay is that the service class was delayed because of non-paging DASD I/O activity.

The SRM collects I/O using and delay information beginning with OS/390 Release 3. Prior to OS/390 Release, any I/O delay is reflected in the UNKNOWN category of delay, and CPExpert will analyze the I/O delay as discussed in Rule WLM350.

The non-paging DASD I/O using and delay information is reported in SMF Type 72 records for each service class period. CPExpert analyzes the non-paging DASD I/O delay (field R723CIOD) for service classes missing their performance goal. CPExpert produces Rule WLM362 when the percent delay caused by non-paging DASD I/O is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE), and a response performance goal has been specified.

After producing Rule WLM362, CPExpert analyzes several possible causes of non-paging DASD I/O delay and reports the result in subsequent rules.

The following example illustrates the output from Rule WLM362:

©Copyright 1997, Computer Management Sciences, Inc.

```
RULE WLM362: NON-PAGING DASD I/O ACTIVITY CAUSED SIGNIFICANT DELAYS

   TSO (Period 1): A significant part of the delay to the service
   class can be attributed to non-paging DASD I/O delay.  The below data
   shows intervals when non-paging DASD delay caused TSO to miss its
   performance goal:

                          AVERAGE
                         DASD I/0  TOTAL DASD  ---AVERAGE DASD I/O TIMES--
   MEASUREMENT INTERVAL  PER TRANS TIME/TRANS  RESP    DISC    CONN   PEND
   10:45-11:00,06MAR1997     64       0.773   0.012   0.005   0.002  0.005
```

Suggestion: From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. The last three of these measures are reported in SMF Type 72 records (fields R723CIWT, R723CIDT, and R723CICT, respectively) for environments prior to OS/390 V2R4. IOSQ time is reported in SMF Type 72 (field R723CIOT) beginning with OS/390 V2R4.

Please refer to the suggestions associated with Rule WLM361 for a discussion of these measures and how to reduce delay in each category.

Rule WLM363: Non-paging DASD wait time was a major cause of DASD delay

Finding: CPExpert has determined that non-paging DASD wait time was a major cause of delay in DASD response for the I/O operations of the service class.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class. The finding applies only with OS/390 Release 3 and subsequent versions.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM361: Non-paging DASD I/O activity caused significant delays

Discussion: Non-paging DASD I/O time is the time from the issuance of the SSCH instruction until the device is selected by the control unit. This time is caused by queuing for the path (wait for channel, wait for control unit or wait for head-of-string), and can be caused by other systems sharing the device (wait for device).

CPExpert examines the non-paging DASD I/O wait time contained in SMF Type 72 records (field R723CIWT). CPExpert produces Rule WLM363 if the non-paging DASD I/O wait time accounted for a significant percent of the non-paging DASD I/O for the service class missing its performance goal.

If the service class missing its performance goal is a transaction service class (for example, composed of CICS 4.1 transactions), CPExpert will identify the server service class (for example, the CICS region). CPExpert will then analyze the DASD I/O times for the server.

The following example illustrates the output from Rule WLM363:

```
RULE WLM363: NON-PAGING DASD WAIT TIME WAS A MAJOR CAUSE OF DASD DELAYS

   TSO: A major part of the DASD I/O delay to the service class
   is attributed to non-paging DASD wait (DASD PEND time and control unit
   queue time).  Please refer to the WLM Component User Manual for advice
   on how to minimize DASD wait time.
```

Suggestion: Large device PEND times usually involve the following situations:

©Copyright 1997, Computer Management Sciences, Inc.

- **Shared devices**. If the device is shared with another system, PEND time may indicate contention with the other system. Large PEND times in shared-device environments usually involve situations very similar to those described under IOSQ time:

  - **Multiple data sets active on the volume**. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing at the channel level (reflected as PEND time) for the single volume.

    If some of the data sets are not required to be shared, then the Data Base Administrator has complete flexibility to move these data sets (subject, of course, to the performance implications of the target devices). These data sets should be moved to a non-shared device.

    If the data sets are required to be shared, then they must be relocated to shared devices.

  - **Multiple applications or users using the same data set on the volume**. Depending upon the data set characteristics, duplicate copies of the data set may be placed on different volumes. This would solve the PEND problems cause by contending systems. If this option is feasible, the data sets could be placed on non-shared devices, likely resulting in even more performance improvement.

  - **Multiple application systems may be using the volume experiencing high PEND times**. In this case, perhaps application redesign or scheduling can solve the problem.

  Additionally, large PEND times for shared devices could be caused by RESERVE from the other system. The applications issuing the RESERVE should be examined to determine whether the RESERVE is required. If the RESERVE **is** required, the above situations should reviewed to determine whether improvements can be achieved.

- **Non-shared devices**. Large PEND times for devices that are not shared may mean that there are insufficient paths available to the device. Too much I/O may be directed to many devices on the path, control unit, or head-of-string. The data sets can be redistributed among different volumes on different paths, control units, or heads-of-string. This will reduce the hardware-level queuing. Alternatively, the entire volume may be moved to a different (less busy) head-of-string or path.

  If redistributing the data sets or moving the volume is not feasible, then the device should have more paths. Depending upon the existing

configuration, this may involve re-configuring existing channel paths, or acquiring additional hardware.

- **Devices attached to cached controllers**. Large PEND times for devices attached to cached controllers may imply a high percent of read miss operations, or non-volatile storage (NVS) writes for IBM-3990-3 devices. Fairchild [1] lists four ways in which staging in caching controllers can cause hidden device busy (with the device busy potentially reflected in high PEND time):

  - The normal (random) caching algorithm stages all records to the end of the track after a requested record is read.

  - The normal (random) caching algorithm stages all records from the beginning of the track to the requested record if a front-end miss occurs.

  - Most writes to extended function IBM-3990 (Model 3) go into NVS with a subsequent destaging required.

  - The sequential caching algorithm stages all records to the end of the track after the requested record is read, and stages in all of the next track. IBM-3990 (Model 3) controllers stages in all of the next three tracks.

- **Dual Copy Initialize**. Large PEND times for IBM-3390 devices may be caused by dual copy initialize. In this case, the dual copy initialize should be turned off.

[1] Fairchild, Bill, "The Anatomy of an I/O Request", *Conference Proceedings*, CMG'90, the Computer Measurement Group, Chicago, IL.

©Copyright 1997, Computer Management Sciences, Inc.

Rule WLM364: Non-paging DASD connect time was a major cause of DASD delay

Finding: CPExpert has determined that non-paging DASD connect time was a major cause of delay in DASD response for the I/O operations of the service class.

Impact: This finding may have a LOW IMPACT or MEDIUM IMPACT on the performance of the device. This finding applies only with OS/390 Release 3 and subsequent versions.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM361: Non-paging DASD I/O activity caused significant delays

Discussion: Connect time is the time in which the device is actually connected to the path. This time includes the data transfer time, but also includes protocol exchange (or "hand shaking") between the various components at several stages of the I/O operation.

The data transfer time obviously is a function of the amount of data being transferred. This simply is the number of bytes transferred divided by the transfer speed (for example, if 4096 bytes were transferred from an IBM-3380 with a transfer speed of 3,000,000 bytes per second, the 4096 bytes would require 4096/3,000,000 seconds; or about 1.36 milliseconds).

Large connect times generally are caused by the following situations:

- A large average block size. This situation may be highly desirable for sequential data sets, but would be undesirable for randomly accessed data.

- Long multi-track searches. For example, the catalog must be searched for cataloged files, the Volume Table of Contents (VTOC) must searched to find a requested file, a directory must be searched for partitioned data sets, etc.. These searches will result in long connect times for the volume involved.

- Program loading from system packs.

CPExpert examines the non-paging DASD I/O connect time contained in SMF Type 72 records (field R723CICT). CPExpert produces Rule WLM364 if the average connect time accounted for a significant percent of

the I/O time for transactions in the service class missing its performance goal.

If the service class missing its performance goal is a transaction service class (for example, composed of CICS 4.1 transactions), CPExpert will identify the server service class (for example, the CICS region). CPExpert will then analyze the DASD I/O times for the server.

The following example illustrates the output from Rule WLM364:

```
RULE WLM364: NON-PAGING DASD CONNECT TIME WAS A MAJOR CAUSE OF DELAYS

  DB2HIGH: A major part of the delay to the service class was due to
  non-paging DASD device connect (CONN) time.  Connect time is caused
  primarily by data transfer.  Please refer to the WLM Component User
  Manual for advice on how to minimize device connect time.
```

Suggestion: As mentioned above, large connect times may be acceptable, depending upon the nature of the application files and why the large connect times occur.

CPExpert suggests that you review the files accessed by the service class missing its performance goal. Based on this review, you can decide whether the large connect times are appropriate or whether action should be take with respect to the application files.

- If the large connect times are appropriate, you may wish to review the performance goal specified for the service class. Depending upon how much connect time was responsible for the I/O delay (and how much the I/O delay accounted for the service class missing its performance goal), you may wish to adjust the performance goal. This, of course, is the easiest solution: you simply adjust the performance goal considering the data transfer requirements of the applications.

- If the large connect times are not appropriate (or if you cannot adjust the performance goal because of management decisions), you may be required to address the application and its files. This step may require considerable effort, depending upon the application, and normally will not be taken lightly.

Rule WLM365: Non-paging DASD disconnect time was a major cause of DASD delay

Finding: CPExpert has determined that non-paging DASD disconnect (DISC) time was a major cause of delay in DASD response for the I/O operations of the service class.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class. This finding applies only with OS/390 Release 3 and subsequent versions.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM361: Non-paging DASD I/O activity caused significant delays

Discussion: DISC time is the time (1) from when the controller initiates a SEEK Channel Command Word (and the seek requires an arm movement) on the device until the SEEK command is complete, (2) plus the time of the rotational delay while the SET SECTOR Channel Command Word is executing, and (3) plus the rotational position sensing (RPS) delay time required because of missed RPS reconnect. Please see Rule WLM350 for a discussion of DISC time.

CPExpert examines the non-paging DASD I/O disconnect time contained in SMF Type 72 records (field R723CIDT). CPExpert produces Rule WLM365 if the average DISC time accounted for a significant percent of the I/O time for transactions in the service class missing its performance goal.

If the service class missing its performance goal is a transaction service class (for example, composed of CICS 4.1 transactions), CPExpert will identify the server service class (for example, the CICS region). CPExpert will then analyze the DASD I/O times for the server.

The following example illustrates the output from Rule WLM365:

```
RULE WLM365: NON-PAGING DASD DISCONNECT TIME WAS A MAJOR CAUSE OF DELAYS

   CICSDEFA: A major part of the delay to the SYSSTC server was due to non-
   paging DASD device disconnect (DISC) time. Disconnect time is caused
   by normal rotation delay (latency), seeking, or missed rotational
   position reconnect (RPS).  Please refer to the WLM Component User Manual
   for advice on how to minimize device disconnect time.
```

Suggestion: Large DISC time is caused by device seek delay, device rotational delay, or device missed RPS reconnect delay.

- **Seek delay**. The SEEK command is responsible for positioning the arm to the proper cylinder. If no positioning is required (that is, the arm is already at the proper cylinder), the device is not disconnected. Seek operations occur because of accessing patterns with data sets and because of accessing patterns between data sets. Large seek times usually involve the following situations:

  - Multiple data sets being active on the volume. The data sets can be redistributed among different volumes, to eliminate the seeking on the single volume.

  - Multiple users using the same data set on the volume. While only one data set is involved, the user or application accessing patterns may require frequent arm movement. A partitioned data set in which several TSO users reference different members is a common situation.

    Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the seeking problems.

- **Rotational delay**. The SET SECTOR command is responsible for locating the proper sector on the track as the disk rotates. (Actually, the SET SECTOR command locates a sector three sectors preceding the desired sector. This sector is called the *angular sector*.) The device is disconnected during the SET SECTOR command operation.

  The rotational delay may be from zero, to the total time required to rotate the disk to the required sector. It is possible that the required sector will be immediately under the head. In this case there is zero rotational delay. On the other hand, the sector could have just passed under the head before the SET SECTOR command was received by the drive. In this case, a full rotation must be accomplished before the required sector is located.

  On average, one-half of the rotation time will be required to locate the sector. This time is referred to as the **average latency** of the device. For example, IBM-3390 devices rotate every 14.1 milliseconds and the average latency is 7.1 milliseconds[1].

[1]These values apply only to Model 1, Model 2, and Model 3. IBM-3390 Model 9 devices rotate every 45.1 milliseconds and the average latency is 22.8 milliseconds.

It is important to realize that the latency is an average based upon many SET SECTOR commands. Any particular SET SECTOR command may have a latency ranging from zero to the maximum rotational delay.

If there are few I/O commands for a particular device in a given measurement interval, it is uncertain what the average latency will be. However, if there are many I/O commands for a particular device in a given measurement interval, the average latency will normally be one-half of the rotational delay.

The average latency may be (and should be) quite small with cached devices. This is because many I/O requests should be satisfied from the cache and have no latency.

If the average DISC per I/O operation is approximately equal to the average latency for the device type, the device would benefit from caching. If the average DISC per I/O operation is much larger than the average latency for the device type, it is unclear whether the device would benefit from caching.

- **Missed RPS reconnect**. The device attempts to reconnect to the path when the angular sector is reached (the angular sector is described above). If the reconnect attempt is successful before the desired sector is reached, then the device connects and the read or write operation can proceed.

  A path must be available for the device to reconnect successfully. If the reconnect is not successful before the desired sector is reached, then the device does not connect, and a complete revolution of the track must occur before the angular sector is again reached. This is called a *missed rotational position sensing reconnect (or missed RPS reconnect)* delay.

  There is no action which can alleviate the initial rotational positioning delay (aside from changing device characteristics, such as implementing caching or buffering at the device level). Over a large number of I/O operations, this initial delay will be one-half the rotation times.

  However, the missed RPS delay is a function of the probability that the path will be busy when the device attempts to reconnect; the busier the path(s), the more missed RPS delay. (Note that the path busy time is a function of the connect time of other actuators. The path cannot be busy from the device itself when the device attempts to reconnect.).

Rule WLM366: Non-paging DASD IOSQ time was a major cause of DASD delay

Finding: CPExpert has determined that queuing in the I/O Supervisor (IOSQ) for non-paging DASD was a major cause of delay in DASD response for the I/O operations of the service class.

Impact: This finding may have a MEDIUM IMPACT or HIGH IMPACT on the performance of the service class. This finding applies only with OS/390 Version 2 Release 4 and subsequent versions.

Logic flow: The following rule causes this rule to be invoked:
Rule WLM361: Non-paging DASD I/O activity caused significant delays

Discussion: IOSQ time is the time from the issuance of a STARTIO macro until the Start SubChannel (SSCH) instruction is issued. After the STARTIO macro is issued, the software determines whether the device is busy. If the device is not busy with this system, the SSCH instruction is issued. However, if the device is busy with this system, the I/O request is queued. Thus, IOSQ time always means that the device is unable to handle additional requests from this system.

Some small IOSQ time is often unavoidable. However, large IOSQ time imply a situation that should be examined. Large IOSQ times result from (1) too many I/O operations directed to the device or (2) lengthy device response times (perhaps caused by high seeking, high RPS delays, or high PEND time).

The following example illustrates the output from Rule WLM366:

```
RULE WLM366: NON-PAGING DASD IOSQ TIME WAS A MAJOR CAUSE OF DELAYS

   BATCHMED: A major part of the delay to the service class was due queuing
   in the I/O Supervisor (IOSQ) for non-paging DASD devices. IOSQ time is the
   time from the issuance of a STARTIO macro until the Start SubChannel (SSCH)
   instruction is issued. Please refer to the WLM Component User Manual
   for advice on how to minimize device IOSQ time.
```

Suggestion: Large IOSQ times usually involve the following situations:

- Multiple data sets may be active on the volume. This situation is the most common and easiest to solve. The data sets can be redistributed among different volumes, to eliminate the queuing for the single volume.

- Multiple users may be using the same data set on the volume. Depending upon the data set characteristics, duplicate copies of the data set placed on different volumes may solve the IOSQ problems.

- Multiple application systems may be using the volume experiencing high IOSQ times. In this case, perhaps application redesign or scheduling can solve the problem.

- A particular application (or system function) may be executing I/O to the device faster than the device can respond.

- The overall device response time (PEND, DISC, and CONN) times may be large, such that the device is unable to provide quick response to the I/O requests. This situation will be revealed by large values in the PEND, DISC, or CONN measures.

Depending on the amount of IOSQ time involved, on budget considerations, and on the business importance of the work being delayed, you might consider acquiring Parallel Access Volumes (PAV). The PAV design tends to eliminate IOSQ time.

Rule WLM370: Non- DASD I/O activity or delay was a major part of execution

Finding: Non-DASD I/O activity or delay experienced by the service class caused significant delays to the service class.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of non-DASD I/O activity and the delay to the service class caused by the non-DASD I/O activity.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal

Discussion: When CPExpert detects that a service class did not achieve its performance goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of delay is that the service class was delayed because of non-DASD I/O activity.

Prior to OS/390 Release 3, CPExpert cannot tell from the Type 72 information whether the I/O operations were directed to tape, to DASD, or to other device types. Prior to OS/390 Release 3, any I/O delay is reflected in the UNKNOWN category of delay, and CPExpert will analyze the I/O delay as discussed in Rule WLM350. However, DASD normally is the fastest medium. If the I/O had been directed to DASD, the delay normally would be less than if the I/O had been directed to other activity. Prior to OS/390 Release 3, CPExpert simply makes an assumption that all I/O activity had been directed to DASD, simply to get a "feel" as to whether the I/O activity could be a significant cause for delay.

The SRM began collecting non-paging DASD I/O using and delay information, and collecting non-DASD I/O using and delay information beginning with OS/390 Release 3.

- Rule WLM361 analyzes non-paging DASD I/O using and delay information.

- This rule (Rule 370) analyzes non-DASD I/O using and delay information.

The non-DASD I/O using and delay information is reported in SMF Type 72 records for each service class period, as a single variable (R723CNDI).

©Copyright 1997, Computer Management Sciences, Inc.

R723CNDI contains a count of samples in which an address space was using non-DASD I/O or was delayed because of non-DASD I/O. The SRM examines each address space or enclave, and adds a sample count for each non-DASD I/O request queued in IOS or active per address space or enclave.

Since the using and delay are combined into a single variable it is not possible to distinguish between non-DASD I/O using and non-DASD I/O delay. However non-DASD using and delay can be a significant part of the I/O activity of many service classes.

When CPExpert detects that a service class misses its response or execution velocity goal, CPExpert computes the non-DASD I/O activity as a percent of the total samples from address spaces or enclaves (R723CNDI/R723CSAC). Since non-DASD I/O activity can not occur when an address space or enclave is idle, CPExpert adjusts the resulting value by the percent of Idle samples (R723CIDL as a percent of all samples). The result is the average number of non-DASD I/O requests queued in IOS or active per address space or enclave. Note that this number can be greater than 100% if an average of more than one non-DASD I/O request was queued in IOS or active.

CPExpert produces Rule WLM370 when the percent by non-DASD I/O is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE).

The non-DASD I/O using and delay are not under the control of the Workload Manager and are not considered in computing or analyzing service class performance. However, a significant I/O delay may be important from the overall performance of the service class. Consequently, CPExpert reports the non-DASD I/O activity.

The following example illustrates the output from Rule WLM370:

```
RULE WLM370: NON-DASD I/O USING OR DELAY WAS A MAJOR PART OF EXECUTION

   BATCHLOW: Non-DASD I/O using or non-DASD I/O delay was a major part of
   the execution time of BATCHLOW (Period 1).  Only the total of non-DASD
   I/O using and delay samples is provided by SMF and there is no way to
   determine whether the non-DASD I/O really caused the service class to
   miss its goal.  However, the non-DASD I/O was a significant part of the
   execution time of the service class during the below intervals.  The
   percentages are computed as a function of the EXECUTION samples on the
   local system (the percentages are adjusted to eliminate IDLE time, to
   reflect the effect when the service class was actually executing).
   Values greater than 100% indicate that an average of more than one
   I/O operation was active concurrently during the execution time.

                                NON-PAGING DASD PCT       NON-DASD PCT
   MEASUREMENT INTERVAL          USING       DELAY      USING AND DELAY
    9:00- 9:15,19NOV1998          47.7         7.7            27.7
   11:00-11:15,19NOV1998          41.1         9.2            18.3
   12:00-12:15,19NOV1998          33.4         6.5            10.2
   14:00-14:15,19NOV1998          32.2         5.1            46.9
   14:15-14:30,19NOV1998          19.9         3.5            43.5
   14:30-14:45,19NOV1998          20.8         1.2            25.0
   16:15-16:30,19NOV1998          42.7         2.6            33.3
   17:00-17:15,19NOV1998          40.6        12.5            12.7
```

Suggestion: Please note that the non-DASD I/O activity did not directly cause the service class to miss its performance goal. However, the non-DASD I/O time was significant, and could have caused overall performance to be degraded.

Rule WLM371: Non-paging DASD I/O activity caused significant delays

Finding: Non-paging DASD I/O activity experienced by the service class caused significant delays to the service class.

This finding applies only to service classes with execution velocity goals, and then applies **only** if I/O using and I/O delay are **not** included in the execution velocity calculation.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT, depending upon the amount of non-paging DASD I/O activity and the delay to the service class caused by the non-paging DASD I/O activity.

Logic flow: The following rules cause this rule to be invoked:
Rule WLM103: Service Class did not achieve execution velocity goal

Discussion: When CPExpert detects that a service class did not achieve its execution goal, CPExpert analyzes the basic causes (see the discussion in the above predecessor rules). One of the possible causes of delay is that the service class was delayed because of non-paging DASD I/O activity.

The SRM collects I/O using and delay information beginning with OS/390 Release 3. Prior to OS/390 Release 3, any I/O delay is reflected in the UNKNOWN category of delay, and CPExpert will analyze the I/O delay as discussed in Rule WLM350.

The non-paging DASD I/O using and delay information is reported in SMF Type 72 records for each service class period. CPExpert analyzes the non-paging DASD I/O delay (field R723CIOD) for service classes missing their performance goal. CPExpert produces Rule WLM371 when the percent delay caused by non-paging DASD I/O is greater than the **WLMSIG** guidance variable in USOURCE(WLMGUIDE), and an execution velocity goal has been specified..

From the perspective of Rule WLM371, I/O using and I/O delay are not considered in computing execution velocity. However, a significant I/O delay may be important from the overall performance of the service class. This is because service classes with an execution velocity goal often have significant I/O activity.

The following example illustrates the output from Rule WLM371:

```
RULE WLM371: NON-PAGING DASD I/O EXPERIENCED SIGNIFICANT DELAYS

   BATCHHI: Non-paging DASD I/O operations experienced significant delay
   during the time that the BATCHHI service class was executing.  The
   percentages are computed as a function of the EXECUTION samples on the
   local system (the percentages are adjusted to eliminate IDLE time, to
   reflect the effect when the service class was actually executing).
   Values greater than 100% for the PCT DELAY indicate that an average of
   more than one DASD I/O operation was delayed concurrently during the
   execution time.

                          AVG DASD   PCT    ---AVERAGE DASD I/O TIMES---
   MEASUREMENT INTERVAL   I/O RATE  DELAY   RESP  IOSQ  WAIT  DISC  CONN
   21:00-21:15,19NOV1998      380    22.7  0.016 0.010 0.001 0.004 0.001
   22:30-22:45,19NOV1998      686    30.6  0.015 0.007 0.001 0.005 0.001
   22:45-23:00,19NOV1998      575    12.5  0.010 0.003 0.001 0.005 0.001
```

Suggestion: Please note that the non-paging DASD I/O activity did not directly cause the service class to miss its execution velocity goal, since non-paging DASD I/O activity was not a part of the execution velocity calculation. However, the non-paging DASD I/O time was significant, and could have caused overall performance to be degraded.

From a high-level view, there are four key measures of DASD performance: IOS Queue (IOSQ) time, pending (PEND) time, disconnect (DISC) time, and connect (CONN) time. The last three of these measures are reported in SMF Type 72 records (fields R723CIWT, R723CIDT, and R723CICT, respectively) for environments prior to OS/390 V2R4. IOSQ time is reported in SMF Type 72 (field R723CIOT) beginning with OS/390 V2R4.

Please refer to the suggestions associated with Rule WLM361 for a discussion of these measures and how to reduce delay in each category.

Rule WLM385: SMF Type 30 (Interval) data was not available

Finding: The service class which missed its performance goal was delayed for an UNKNOWN delay reason. CPExpert attempted to estimate the amount of UNKNOWN delay that was related to DASD I/O. However, SMF Type 30 (Interval) data was not available for the service class.

Impact: This finding has NO IMPACT. The finding is produced for information purposes to explain that CPExpert was unable to estimate the amount of DASD I/O using SMF Type 30 (Interval) data. This finding will not apply with OS/390 Release 3 and subsequent releases of MVS, as non-paging DASD I/O delay is reported beginning with OS/390 Release 3.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300:	Service Class was delayed for UNKNOWN delay
Rule WLM301:	Server Service Class was delayed for UNKNOWN delay

Discussion: As described in the above rules, the UNKNOWN category of workload delay means that the Workload Manager was unable to identity the cause of the delay. The UNKNOWN delay normally is caused by something over which the System Resources Manager has no control. The IBM documentation explains that this delay category potentially "includes I/O delay, ENQ delay, etc." No information is available about other causes of UNKNOWN delay; the UNKNOWN delay is simply a category of delay which the SRM cannot identify.

CPExpert has detected that a service class missed its performance goal, and experienced significant I/O delays. The **TYPE30_V** guidance variable in USOURCE(GENGUIDE) was set to "Y" to indicate that SMF Type 30 Interval Recording was turned on and the **TYPE30DD** guidance variable in USOURCE(GENGUIDE) was set to "Y" to indicate that the modification to MXG or MICS had been implemented to collect detailed DASD information.

CPExpert determined that a service class missed its performance goal, and CPExpert attempted to analyze SMF Type 30(Interval) data related to the service class. SMF Type 30(Interval records **were** available for the service class for some RMF intervals. However, CPExpert could not find any interval records for the service class which had missed its performance goal, **during the interval(s) when the goal was missed**. CPExpert produces Rule WLM385 to alert you to this situation.

©Copyright 1997, Computer Management Sciences, Inc.

The following example illustrates the output from Rule WLM385:

```
RULE WLM385: SMF TYPE 30(INTERVAL) DATA WAS NOT AVAILABLE

  TSO: SMF interval recording (for SMF Type 30 data) was turned on for
  the service class, but the interval records were not available during
  the following measurement intervals.  Consequently, CPExpert cannot
  evaluate whether the large UNKNOWN delay for TSO (Period 1) could
  have been caused by DASD I/O.  However, the SMF Type 72 records did
  reflect I/O activity for Service Class TSO (Period 1), as shown
  below.  This I/O activity COULD have caused TSO to miss its response
  goal, but CPExpert does not have sufficient information on which to base
  such a conclusion.
                                SMF TYPE 72      TOTAL    AVG  EXCP
 MEASUREMENT INTERVAL         TOTAL EXCP COUNT   TRANS    PER TRANS
 10:45-11:00,06MAR1995               3582          199        18
```

Suggestion: This finding is produced for information purposes, to let you know that a substantial amount of the service class delay was not accounted for by CPExpert's analysis.

Two alternatives should be considered:

- **The CPExpert code has an error**. The code that analyzes the SMF Type 30 records and correlates the result with SMF Type 72 information is relatively complex. It is possible that an error exists in this code. If Rule WLM385 should be produced, please call Computer Management Sciences at (703) 922-7027 .

- **The SMF Type 30 (Interval) records were temporarily turned off**. Rule WLM090 will be produced if no the SMF Type 30 (Interval) records at all were found for the service class missing its goal. Rule WLM385 deals with the situation where some SMF Type 30 (Interval) records were found in the performance data base, but not for the RMF intervals in which the service class missed its goal. An operations situation may exist if interval recording was temporarily turned off.

Rule WLM390: UNKNOWN Delay was not accounted for by above analysis

Finding: The service class which missed its performance goal was delayed for an UNKNOWN delay reason. CPExpert attempted to identify the components of the UNKNOWN delay, but a significant amount of UNKNOWN delay remained after the estimated values were subtracted from the UNKNOWN delay reported by SMF.

Impact: This finding has NO IMPACT. The finding is produced for information purposes to explain that not all UNKNOWN delay was estimated.

Logic flow: The following rules cause this rule to be invoked:

Rule WLM300: Service Class was delayed for UNKNOWN delay
Rule WLM301: Server Service Class was delayed for UNKNOWN delay

Discussion: As described in the above rules, the UNKNOWN category of workload delay means that the Workload Manager was unable to identity the cause of the delay. The UNKNOWN delay normally is caused by something over which the System Resources Manager has no control. The IBM documentation explains that this delay category potentially "includes I/O delay, ENQ delay, etc." No information is available about other causes of UNKNOWN delay; the UNKNOWN delay is simply a category of delay which the SRM cannot identify.

In many environments, the UNKNOWN delay will consist of I/O delays. Consequently, CPExpert estimates potential I/O delays, as described in Rule WLM350, Rule WLM351, and Rule WLM352. These result from these rules is subtracted from the UNKNOWN category of delay reported in SMF Type 72 records.

CPExpert produces Rule WLM390 when the remaining UNKNOWN delay accounts for a significant amount of the delay to the service class. The purpose of Rule WLM390 is simply to alert you to the fact that CPExpert's estimated delays did not account for all of the UNKNOWN delay category.

In early execution of the WLM Component of CPExpert, the UNKNOWN delay category often accounted for a significant amount of total delay, and Rule WLM390 was regularly produced. Perhaps with increased user experience with executing MVS/ESA SP5(Goal Mode) and increased experience executing the CPExpert WLM Component, additional information will become available about the UNKNOWN category of delay.

©Copyright 1994, Computer Management Sciences, Inc.

We should be able to further identify the components of UNKNOWN delay as more information becomes available.

The following example illustrates the output from Rule WLM390:

```
RULE WLM390: UNKNOWN DELAY WAS NOT ACCOUNTED FOR BY ABOVE ANALYSIS

   The UNKNOWN delay causing Service Class ST_USER (Period 1) to miss
   its performance goal was not accounted for by CPExpert's analysis.
   The UNKNOWN delay could have been caused by address spaces waiting
   for action by another service class, by enqueues, by waiting for I/O
   operations not reflected by the DASD analysis, etc.  There is not
   enough information to identify the cause of the UNKNOWN delay to
   ST_USER in the following measurement intervals:

                             AVERAGE    AVERAGE UNKNOWN     UNKNOWN DELAY
   MEASUREMENT INTERVAL     RESPONSE    DELAY PER TRANS    UNACCOUNTED FOR
   14:00-14:15,01MAR1994     10.770          7.314              6.991
   14:15-14:30,01MAR1994     14.992         10.777              7.127
   14:30-14:45,01MAR1994     19.445         14.419              6.248
   14:45-15:00,01MAR1994      3.849          2.708              1.618
   15:00-15:16,01MAR1994     18.112         11.882             11.882
```

Suggestion: This finding is produced simply for information purposes, to let you know that a substantial amount of the service class delay was not accounted for by CPExpert's analysis.

Rule WLM400: Page-in from auxiliary storage was a major cause of delay

Finding: CPExpert has determined that waiting for page-in from auxiliary storage was a major cause of the service class not achieving its performance goal.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The impact of this finding depends upon the percent of time transactions in the service class were waiting for pages from auxiliary storage. A high percent waiting for pages means HIGH IMPACT while a low percent waiting for pages means LOW IMPACT.

Please note that the percentages reported by CPExpert are computed as a function of **the active time of the transactions**, rather than percentages of RMF measurement interval time. The percentages show the impact of page-in delay **on the transactions**, rather than the impact of page-in from an overall system view. This data presentation approach is significant when the service class being delayed is a **server** service class; the page-in delays represent delays to the response times of the served transaction!

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM150:	Server Service Class delays
Rule WLM151:	Server Service Class delays

Discussion: The MVS virtual storage environment operates on the principles that:

- The central storage required by any particular address space during execution is a subset of the total central storage required to load the address space. Much of the storage required to load an individual address space is often unused. This storage that **is** regularly used is referred to as the "working set" of the address space. The working set is typically a small part of the overall central storage requirement to initially load an address space. The remaining (typically large) amount

of central storage can be used by other address spaces loaded concurrently.

- Idle central storage should be used to prevent unnecessary I/O operations. In fact, central storage generally should be managed to maximize the use of central storage while minimizing I/O operations.

- Central storage can be "allocated" to address spaces based upon the importance of the address space. That is, the "working set" of a low priority address space can be constrained if necessary to allow more important address spaces to have adequate central storage.

- With appropriate external storage and process controls, users of a virtual environment should notice little difference between the performance of the virtual environment and the performance of a non-virtual environment.

Exchanging pages between central storage and auxiliary storage (and between central storage and expanded storage if expanded storage is installed) is the way MVS allows multiple address spaces to concurrently use a finite amount of central storage. When an address space requires a page of storage that has been removed from central storage, a "page fault" occurs. The address space (actually, the TCB or SRB associated with the address space) is unable to continue processing until the page fault is resolved. MVS will locate the page and bring it into central storage.

- The page might actually be in central storage waiting a page-out operation. These pages are "reclaimed" and made available without further effort. No statistics are maintained on the number of reclaimed pages, but this number normally should be small.

- The page may be in expanded storage (for systems with expanded storage). These pages are moved directly from expanded storage; the page-in time is very small (various studies have reported page-in times from expanded storage in the range of 40-75 microseconds). Delays for these page operations do not generally cause a performance problem[1].

- If the page is not in central or expanded storage, the page must be physically brought in from auxiliary storage. It is these page-in operations which Rule WLM400 addresses.

If the page is in expanded storage or in auxiliary storage, a page frame in central storage must be available to hold the page being paged in. The

[1]While delays for page-in operations from expanded storage does not normally cause problems, there are some situations in which the page-in rate from expanded storage can seriously degrade performance. The Workload Manager will monitor and potentially manage service classes or address spaces which experience or cause a high paging rate from expanded storage.

Real Storage Manager normally maintains a number of "available" page frames in central storage to accommodate the page.

The time from the page fault until the required page is available is considered page delay time. During this time, the address space requiring the page normally must wait. During the waiting time, the central storage associated with the address space is wasted for the page delay time. Additionally, other resources allocated to the address space are unusable during the page fault resolution time.

The page delay time may have other, potentially more serious, implications. If the address space is associated with a response-critical application (e.g., a TSO trivial transaction), end-user response will be delayed for the time required to resolve the page fault. If many page faults occur, response may degrade to less than the performance goals for the service class.

The SMF Type 72 records contain information which can be analyzed to determine the amount of delay a service class experienced as a result of page-in operations from auxiliary storage. The page-in delay from auxiliary storage is separately reported in the following delay categories:

- **Private area page-in from auxiliary storage delay**. This delay category means that the address space was experiencing page faults in the private area and the pages were coming from auxiliary storage.

- **Common area page-in from auxiliary storage delay**. This delay category means that the address space was experiencing page faults in the Common area and the pages were coming from auxiliary storage.

- **Cross-memory page-in from auxiliary storage delay**. This delay category means that the address space was experiencing page faults in cross-memory access and the pages were coming from auxiliary storage.

- **VIO page-in from auxiliary storage delay**. This delay category means that the address space was experiencing page faults in VIO and the pages were coming from auxiliary storage.

- **Standard hiperspace page-in from auxiliary storage delay**. This delay category means that the address space was experiencing page faults in standard hiperspace and the pages were coming from auxiliary storage.

- **ESO hiperspace page-in from auxiliary storage delay**. IBM has defined this state to mean that the address space was experiencing page faults in ESO hiperspace and the pages were coming from auxiliary

storage. Pages in ESO hiperspace are, by definition, resident only in expanded storage (ESO = Expanded Storage Only), and are never migrated to auxiliary storage. IBM offers the following explanation[2]:

"The execution delay for ESO hiperspaces is a calculated value based on the assumption that if an application does a read for an ESO hiperspace page and that page is no longer available (has been cast out), the application will read the data from DASD somewhere. WLM/SRM takes the number of times a read failed in this way and multiplies it by the number of delay samples we expect a read of a page from DASD to represent and report the product as the execution delay samples for ESO hiperspace. This obviously is not a perfect solution, but we needed some way to get an estimate of how much delay is caused to an address space by not having enough expanded for an ESO hiperspace. Such an estimated is needed to properly manage the amount of expanded owned by the address space to the address space's goal."

CPExpert sums the page-in delays for all delay categories. CPExpert produces Rule WLM400 if the total page-in delay from auxiliary storage was a major reason the service class identified in the predecessor rules did not meet its performance goal.

The following example illustrates the output from Rule WLM400:

```
RULE WLM400: PAGE-IN FROM AUXILIARY STORAGE WAS MAJOR PERFORMANCE PROBLEM

   Page-in from auxiliary storage was a primary or secondary reason BATCH
   (Period 1) missed its performance goal.  Auxiliary storage paging caused
   the following delays to BATCH (Period 1), shown by category of page-in:

                            PERCENT
                            PAGE-IN   --PERCENT DELAY BY PAGE-IN CATEGORY--
   MEASUREMENT INTERVAL      DELAY    PVT   COMM   XMEM   VIO   HIPR    ESO
   15:00-15:16,01MAR1994      9.2     0.0    0.0    9.2   0.0    0.0    0.0
```

Suggestion: Page-in delays can be reduced in two basic ways: (1) reduce the time to resolve page faults and (2) reduce the number of page faults.

If the total page-in from auxiliary storage delay is unacceptable, CPExpert recommends that the following actions be considered:

- **Make sure that the paging configuration is optimal**. Review the recommendations in Section 2 of the MVS Initialization and Tuning Guide. CPExpert may produce rules in the WLM050(series) to identify

[2]IBM TALKLink RMF FORUM appended at 15:39:18 on 95/05/29 GMT (by YOCOM at KGNVMC)
Subject: Workload Activity Report

potential problems in the paging configuration. The most common problem has been that installations allocate too few local page data sets.

- **Review performance goals and importance**. The Workload Manager will attempt to manage system resources (CPU and processor storage) to meet the performance goals of important workloads. You should make sure that the performance goals and importance levels have been properly specified (1) for service classes with more restrictive performance goals or (2) for service classes at higher level or same level goal importance.

- **Reschedule the workload**. Schedule lower priority workloads to a time when they do not compete with critical applications. The Workload Manager will often swap out lower priority workloads to reduce page-in delay for higher priority workloads. However, the Workload Manager may require some elapsed time to identify the problem and take action. Depending upon the dynamics of the workload mix, the Workload Manager may not be as successful as would manual rescheduling.

- **Ignore the finding**. You may decide that the service class experiencing page-in delays from auxiliary storage is insufficiently important to worry about. The BATCH service class in the example output could be an example of this; you might not worry that batch workload periodically experiences page-in delays and the BATCH service class misses its performance goal.

  You can exclude service classes from analysis[3] by CPExpert if this situation occurs regularly and becomes an annoyance.

- **Acquire additional processor storage**. Page faults occur because the required page is not available in central storage. You may be able to reduce page faults by acquiring additional central storage. Alternatively, you may consider acquiring additional expanded storage, since page fault resolution from expanded storage is extremely fast.

  Acquiring additional processor storage might not reduce page-in delays in some environments. Depending upon the nature of the applications, adding additional central or expanded storage might not have a noticeable effect.

- **Acquire faster paging devices**. If the above options have been exhausted and paging delays are still unacceptable, you should consider acquiring faster paging devices.

[3]Use the EXCLUDE guidance in USOURCE(WLMGUIDE) to exclude service classes from analysis.

Rule WLM450: Swap-in was a major cause of delay

Finding: CPExpert has determined that waiting for swap-in was a major cause of the service class not achieving its performance goal.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The impact of this finding depends upon the percent of time transactions in the service class were waiting for address space swap-in. A high percent waiting for swap-in means HIGH IMPACT while a low percent waiting for swap-in means LOW IMPACT.

Please note that the percentages reported by CPExpert are computed as a function of **the active time of the transactions**, rather than percentages of RMF measurement interval time. The percentages show the impact of swap-in delay **on the transactions**, rather than the impact of swap-in from an overall system view. This data presentation approach is significant when the service class being delayed is a **server** service class; the swap-in delays[1] represent delays to the response times of the served transaction!

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM150:	Server Service Class delays
Rule WLM151:	Server Service Class delays

Discussion: The SRM identifies seventeen reasons that address spaces are swapped out; some of the swaps are a natural function of the SRM's design and some swaps are preventable:

- **Terminal Input Wait swaps** occur because transactions are waiting for terminal input. This reason means the SRM has been notified that a TSO session is in terminal wait after issuing a TGET. The SRM verifies

[1]In practice, swap-in delay should rarely occur for server service classes. The address spaces associated with server service classes usually are non-swappable, although some organizations do make test CICS regions swappable. If the address spaces associated with a server service class are non-swappable, the address spaces will not normally incur swap-in delays.

©Copyright 1994, Computer Management Sciences, Inc.

that the address space is in a long wait before completing the swap. Terminal Input Wait swaps are the most common reason for TSO transaction swaps (usually accounting for 80-90% of all TSO swaps). These swaps generally are a function of the user community and its interaction; there usually is no action that can be taken to prevent these swaps.

- **Terminal Output Wait swap** occur because transactions are waiting for terminal output buffers. This reason means the SRM has been notified that a TSO session is in terminal wait after issuing a TPUT. The SRM verifies that the address space is in a long wait before completing the swap. Wait for terminal output buffer is one of the conditions that signals a new transaction. Terminal Output Wait swaps often account for about 10-15% of all TSO swaps. However, with proper values specified in the TSOKEYxx member of SYS1.PARMLIB, these swaps often can be reduced to less than 1%.

- **Long Wait swap** occur because transactions have requested a swap. Transactions request swaps because of some condition (e.g., WAIT,LONG=YES macro issued, an STIMER wait value of 0.5 seconds or more, or ENQs). Long Wait swaps generally account for less than 1% of all TSO swaps. However, the frequency of Long Wait are application-dependent.

  There is no action that should be taken with regard to these swaps; an application is properly advising the SRM that it is entering a protracted wait.

- **Auxiliary Storage Shortage swaps** occur because insufficient page or swap data sets have been defined. Auxiliary Storage Shortage swaps are very serious. It is unlikely that these swaps occur.

- **Real Pageable Storage Shortage swaps** occur because the Real Storage Manager is unable obtain real memory pages for the Available Frame Queue. Real Pageable Storage Shortage swaps are very serious. It is unlikely that these swaps occur often.

- **Detected Wait swaps** occur because the SRM detects that a resident transaction has not been dispatchable for two seconds of real time or eight SRM seconds, without issuing the WAIT,LONG=YES macro. Detected Wait swaps usually are caused by cross memory services, applications that treats the terminal as SYSIN or SYSPRINT, teleprocessing applications (e.g., test CICS regions) that are not marked non-swappable, etc.

Additionally, STIMER wait values of less than the 0.5 seconds required to trigger a Long Wait swap may trigger a Detected Wait swap if the wait time is more than 8 SRM seconds.

From this definition of Detected Wait swaps, these swaps generally should be a fairly small percentage of the overall swaps. However, Detected Wait swaps commonly account for almost 5% of the total swaps, and sometimes account for over 30% of the total swaps.

- **Request swaps** occur because V=R or non-swappable was specified in the Program Properties Table, or an authorized program has directed that an address space be swapped out (for example, to terminate the address space). These swaps generally occur very infrequently.

- **Enqueue Exchange swaps** occur in which an address space is swapped out because a user is enqueued on a resource held by a swapped out transaction. For example, these swaps occur when a TSO user requests access to a resource (e.g., a file) held by some other address space (e.g., another TSO user, a batch job, etc.). These swaps often have far more impact than their frequency indicates. This is because the SRM will swap in the holder of the resource and allow the user to remain in storage for some time before the user is eligible for swap out.

- **Exchange on Recommendation Value swaps** occur when the SRM has determined that a swapped out transaction in a particular domain[2] is ready to be swapped in, the swapped out transaction has higher "priority" than a transaction in storage in the same domain, and the domain is at its target MPL. Under these conditions, the transactions are "exchanged" in storage. Exchange swaps should rarely occur.

- **Unilateral swaps** occur because the SRM has determined that the number of address spaces in storage for a domain is larger than the target MPL for the domain.

- **Transition to Non-Swappable swaps** - swaps because the transaction becomes non-swappable after being initially swapped in. This happens once for each non-swappable address space, since the SRM doesn't know that the address space is non-swappable until it is swapped in.

- **Improve System Paging Rate** swaps - swaps because the Workload Manager has determined that the system page fault rate exceeds a

[2]Domains are maintained by the SRM in Goal Mode, even though the specification and control of domains has been removed from the user interface. The Workload Manager creates domain control table entries for each service class period so long as the service class period is associated with address spaces (that is, the Workload Manager does not create domain control table entries for service classes representing CICS or IMS transactions).

©Copyright 1994, Computer Management Sciences, Inc.

threshold. This threshold arbitrarily establishes a limit on the number of page faults which the Workload Manager considers acceptable.

- **Improve Central Storage Usage swaps** - swaps because the Workload Manager (1) has decided that too much processor time is spent in resolving page faults, (2) has begun address space monitoring, and (3) has determined that restricting the target working set of a monitored address space did not achieve an acceptable reduction in the "unproductive" CPU time spent resolving page faults. Under these conditions, the Workload Manager will swap out one of the monitored address spaces to improve central storage usage.

- **Make Room to swap in a user who has been swapped out too long** - swaps because the SRM has determined that a user who has been swapped out to improve central storage usage has been swapped out longer than the thresholds (30 seconds for a TSO user and 10 minutes for non-TSO user). If the swapped out user cannot fit into processor storage, the Workload Manager will select an address space to swap out to make room for the swapped out user (effectively performing an Exchange Swap between the two address spaces).

- **APPC Wait swaps** - swaps because the SRM has detected that an address space is waiting for a response in an Advanced Program to Program Communication environment.

- **OMVS input wait swaps** - swaps because the OpenEdition MVS Shell is waiting for input.

- **OMVS output wait swaps** - swaps because the OpenEdition MVS Shell is waiting for output to be complete.

The Workload Manager defines an MPL-in target and MPL-out target for each service class period. The MPL-in target represents the number of address spaces that must be in the swapped-in state for the service class to meet its performance goal. The MPL-out target is the maximum number of address spaces allowed to be in the "swapped-in" state.

Additionally, the Workload Manager defines swap protect time for service class periods. Swap protect time is the time in milliseconds swapped-out address spaces will remain in processor storage before becoming candidates for swap to auxiliary storage. Swap protect time is similar to the "think time" used in previous versions of MVS.

RMF Monitor I provides information on swap activity for the overall system in SMF Type 71 Records (Paging Activity). For each swap type, SMF Type 71 records provide information about whether the swap was physical or

logical, whether it went to auxiliary storage or to expanded storage, etc. Unfortunately, there is no information to associate swap reasons to particular service classes.

Swapping is expensive: swapping requires processor resources and swapping places a load on the paging subsystem. Swapping out ready users incurs the resource expense and delays the users. Additionally, swapped out users retain ownership of their allocated files and may delay other processing.

On the other hand, it is unreasonable to allow system resources to remain temporarily idle while there is work to be done. There is a tradeoff: swapping users versus allowing system resources to remain idle. If the resources actually are to remain idle for an extended period, then it is better to swap other users in and allow them to use the idle resources. The swapping overhead simply involves using resources that otherwise would be unused. If the system becomes active, then the users should not be swapped.

CPExpert produces Rule WLM450 if swap-in was a major reason the service class identified in the predecessor rules did not meet its performance goal.

Suggestion: The Workload Manager (in concert with the System Resources Manager) provides most of the control over swapping. Unlike earlier versions of MVS, users have little direct control and generally cannot specify parameters to directly reduce swapping[3].

Swap-in delays can be reduced in two basic ways: (1) reduce the time to swap in an address space and (2) reduce the number of swaps.

If the swap-in delay is unacceptable, CPExpert recommends that the following actions be considered:

- **Make sure that the paging configuration is optimal**. Review the recommendations in Section 2 of the MVS Initialization and Tuning Guide. CPExpert may produce rules in the WLM050(series) to identify potential problems in the paging configuration. The most common problem has been that installations allocate too few local page data sets.

- **Review performance goals and importance**. The Workload Manager will attempt to manage system resources (CPU and processor storage)

[3]One significant exception to this statement is Terminal Output Wait swaps. Users often can adjust the TSOKEYxx parameters to reduce Terminal Output Wait swaps. Please refer to Rule WLM070 for a discussion of Terminal Output Wait swaps. Additionally, users may be able to reduce Detected Wait swaps. Please refer to Rule WLM071 for a discussion of Detected Wait swaps.

to meet the performance goals of important workloads. You should make sure that the performance goals and importance levels have been properly specified for service classes with more restrictive performance goals or service classes at higher level or same level goal importance.

- **Reschedule the workload**. Schedule lower priority workloads to a time when they do not compete with critical applications. The Workload Manager will often swap out lower priority workloads to reduce page-in delay for higher priority workloads.

  However, the Workload Manager may require some elapsed time to identify the problem and take action. Depending upon the dynamics of the workload mix, the Workload Manager may not be as successful as would manual rescheduling.

- **Ignore the finding**. You may decide that the service class experiencing swap-in delays from auxiliary storage is insufficiently important to worry about. The BATCH service class in the example output could be an example of this; you might not worry that batch workload periodically experiences swap-in delays and the BATCH service class misses its performance goal.

  You can exclude service classes from analysis[4] by CPExpert if this situation occurs regularly and becomes an annoyance.

- **Acquire additional processor storage**. Swap-in of address spaces occurs because the System Resources Manager has swapped address spaces out of processor storage to make page frames available for other address spaces. You may be able to reduce the swap-out of address spaces by acquiring additional central storage.

  Alternatively, you may consider acquiring additional expanded storage, since swap-in from expanded storage is extremely fast.

  Acquiring additional processor storage might not reduce swap-in delays in some environments. Depending upon the nature of the applications, adding additional central or expanded storage might not have a noticeable effect.

- **Acquire faster paging devices**. If the above options have been exhausted and swap-in delays are still unacceptable, you should consider acquiring faster paging devices.

[4]Use the EXCLUDE guidance in USOURCE(WLMGUIDE) to exclude service classes from analysis.

- **Use swap data sets**. This option may be applicable only in a small number of installations; swap data sets are not commonly used. In fact, CPExpert will check for the presence of swap data sets and will produce Rule WLM061 if swap data sets are defined. However, there are unusual circumstances in which swap data sets are appropriate.

  Swap data sets can be used by the Auxiliary Storage Manager (ASM) to contain Local System Queue Area (LSQA) and private area pages that are swapped in with the address space.

  For systems with expanded storage, the RSM and SRM may divide the working set pages into a primary and secondary working set[5].

  - **Primary working set**. The **primary working set** consists of LSQA pages, fixed pages, and one page from each virtual storage segment that is included in the working set[6].

    The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage.

    - The primary working set may be migrated to swap data sets if swap data sets are defined and if sufficient space exists on the swap data sets.

    - If swap data sets are not defined or if insufficient space exists on the swap data sets, the primary working set is migrated to local page data sets.

  - **Secondary working set**. The **secondary working set** consists all working set pages not included in the primary working set. These are most non-LSQA, non-fixed, working set pages. Notice that the secondary working set does not include swap trim pages[7]

  The primary working set may be sent to expanded storage or may migrated from expanded to auxiliary storage. The secondary working set will always be migrated to local page data sets.

---

[5]This division is done only if the swap is to be done to expanded storage. If the swap is to be directly to auxiliary storage, the division is not done (a swap directly to auxiliary storage is called a **single stage swap**).

[6]The working set is composed of those address spaces with UIC of zero or one (and potentially an "enriched" working set with UIC greater than one if storage is not a constraint). A virtual storage segment is one megabyte of virtual storage.

[7]Swap trim pages are those pages trimmed from an address space before it is swapped out. The swap trim pages are the pages in central storage at swap time, which are not included in the working set. The swap trim pages may be sent to expanded if they meet the expanded storage criteria or they will be sent to auxiliary storage.

There are several advantages to using **only** local page data sets, rather than a mixture of swap data sets and local page data sets.

- The ASM load balancing algorithm selects the local page data set with the best performance to receive a page group. This algorithm automatically helps correct performance problems if local page packs are on heavily loaded paths or if local page packs are not dedicated. The ASM does not apply the load balancing algorithm to swap data sets.

- With expanded storage, most of the migration paging (that is, the migration of the secondary working set and migration of swap trim pages) is automatically sent to local page data sets. Thus, most of the pages associated with a swap (either directly in the case of the secondary working set, or indirectly in the case of swap trim pages) will be sent to local page data sets regardless of whether swap data sets are used. Consequently, swap data sets tend to be under-utilized in an expanded storage environment.

- Overall system performance normally would be much better if the volumes which were defined as swap data set volumes were redefined for local page data sets. The local page data sets would individually have a lower average page rate since there would be more volumes available (that is, the paging load would be spread over more volumes).

  For example, suppose you had defined four local page data sets and two swap data sets. Performance would normally be significantly improved if you redefined the swap data sets as local page data sets, for a total of six local page data sets.

That aside, there are circumstances in which you should use swap data sets. For example, you may have very large swap sets in an environment without adequate expanded storage. You may wish to retain swap data sets to prevent critical page-in operations from being slowed by the I/O required to service large swap sets.

The following issues should considered:

- Delay of critical page-in operations is unlikely to exist in an expanded storage environment. Since only the primary working set may be migrated to swap data sets (unless the swap is a single stage swap), little advantage is gained by having swap data sets. That is, the secondary working set will always migrate to local page data sets and the secondary working set is usually significantly larger than the primary working set. Since only the primary working set would be

migrated, only the primary working set would be effected by having swap data sets.

- You normally should have sufficient local page data sets such that the ASM can initiate swap-out I/O operations in parallel to local page data sets. If the I/O operations are initiated in parallel, then the maximum delay to page-in operations normally would be only the time required to transfer a page group (30 pages for local page data sets).

  The time to transfer a page group normally would be about 50-60 milliseconds for an IBM-3380 paging device (the possible seek operation, search operation, and data transfer), and these times would become significantly less if the DASD were cached or if IBM-3390 devices were used for paging. This periodic delay would be offset if the swap data sets were converted to local page data sets, since more local page data sets would result in a lower average page-in time.

- Under some circumstances, the migration rate may be high. If the migration rate is high, one implication is that there are few available pages in expanded storage. (The only purpose of migrating pages is because there is an insufficient number of available expanded storage pages.)

  If there are few available expanded storage pages, the SRM will direct swaps to auxiliary as **single-stage** swaps, and will not prepare a primary and secondary working set. In this situation, allocating swap data sets may prevent the single-stage swaps from overloading the local page data sets.

  Of course, if many swaps are sent to auxiliary rather than to expanded, you have basic problems with your expanded storage environment.

Rule WLM480: Target Multiprogramming Level was a major cause of delay

Finding: CPExpert has determined that the target multiprogramming level (MPL) was a major cause of the service class not achieving its performance goal.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on performance of your computer system. The impact of this finding depends upon the percent of time transactions in the service class were waiting for target MPL before address space swap-in began. A high percent waiting for MPL means HIGH IMPACT while a low percent waiting for MPL means LOW IMPACT.

Please note that the percentages reported by CPExpert are computed as a function of **the active time of the transactions**, rather than percentages of RMF measurement interval time. The percentages show the impact of MPL delay **on the transactions**, rather than the impact of MPL delay from an overall system view. This data presentation approach is significant when the service class being delayed is a **server** service class; the MPL delays[1] represent delays to the response times of the served transaction!

Logic flow: The following rules cause this rule to be invoked:

Rule WLM101:	Service Class did not achieve average response goal
Rule WLM102:	Service Class did not achieve percentile response goal
Rule WLM103:	Service Class did not achieve execution velocity goal
Rule WLM104:	Subsystem Service Class did not achieve average response goal
Rule WLM105:	Subsystem Service Class did not achieve percentile response goal
Rule WLM150:	Server Service Class delays
Rule WLM151:	Server Service Class delays

Discussion: Domains are maintained by the SRM in Goal Mode, even though the specification and control of domains has been removed from the user interface. The Workload Manager creates domain control table entries for each service class period so long as the service class period is associated with address spaces (that is, the Workload Manager does not create domain control table entries for service classes representing CICS or IMS transactions).

[1]In practice, MPL delay should rarely occur for server service classes. The address spaces associated with server service classes usually are non-swappable, although some organizations do make test CICS regions swappable. If the address spaces associated with a server service class are non-swappable, the address spaces will not normally incur MPL delays.

©Copyright 1994, Computer Management Sciences, Inc.

The Workload Manager defines an MPL-in target and MPL-out target for each service class period which is assigned to a domain[2]. The MPL-in target represents the number of address spaces that must be in the swapped-in state for the service class to meet its performance goal. The MPL-out target is the maximum number of address spaces allowed to be in the swapped-in state.

The Workload Manager adjusts the MPL-in target and MPL-out target, if necessary to achieve performance goals. The adjustments are made during the policy adjustment interval.

The Workload Manager adjusts the system-wide target MPL and subsequently may adjust the target MPL for individual service class periods. The Workload Manager adjusts the target MPL for the following reasons:

- **CPU is over-utilized**. The Workload Manager will decrease the system target MPL if the Workload Manager detects that the CPU is over-utilized.

- **CPU is under-utilized**. The Workload Manager will increase the system target MPL if the Workload Manager detects that the CPU is under-utilized.

- **Too much auxiliary storage paging**. The Workload Manager will decrease the system target MPL if the Workload Manager detects that too much auxiliary storage paging occurred.

- **Too much unmanaged paging**. The Workload Manager will decrease the system target MPL if the Workload Manager detects that too much unmanaged paging occurred.

- **Storage shortage below 16 megs**. The Workload Manager will decrease the system target MPL if the Workload Manager detects that there is a shortage of storage below 16 megs.

The above system utilization actions normally will cause the target MPLs for a service class period to be adjusted.

In addition to the system utilization actions, the Workload Manager may take action because a service class period did not meet its performance goal. These actions also are taken at the policy adjustment interval. If a

[2]Actually, the targets are associated with the domain, but it is easier to think of them as being associated with the service class period since domains are no longer part of the user interface

service class period is not meeting its performance goal, the Workload Manager may increase the target MPLs for the service class period. If appropriate, the Workload Manager may concurrently decrease the target MPLs for a service class period at a lower importance.

The IBM *MVS/ESA Programming: Workload Management Services* document (see the "MPL Policy Example" in the “Examples of interpreting SMF Type 99 data” section) provides an excellent example of the Workload Manager's decision process in adjusting the MPLs to meet performance goals.

The Workload Manager may also decrease the MPLs of a service class period if required by working set management. This action would normally be taken only after the working set manager had decided that there were no opportunities to decrease system paging by managing a particular address space.

Additionally, the Workload Manager may decrease or increase the target MPLs for service class periods during "time driven housekeeping". The Workload Manager will implement time driven housekeeping to make periodic adjustments of various resource policies. During time driven housekeeping, the Workload manager may increase or decrease the target MPLs for service class periods based on its analysis of the impact the MPL "slots" have had on the performance of the service class period.

The objectives of these adjustments to the system MPL or the target MPLs of individual service class periods are to (1) allow sufficient workload into the multiprogramming set such that the system is adequately used, (2) exclude workload if necessary to prevent the system from being over-utilized, and (3) make sure that the performance goals of individual service class periods are achieved.

The Workload Manager cannot always achieve these objectives for every service class. The Workload Manage might have to exclude lower importance service classes from the multiprogramming set in order to achieve the performance goals of higher importance service classes.

CPExpert produces Rule WLM480 when the target MPL was a major cause of delay to a service class not achieving its performance goal.

The following example illustrates the output from Rule WLM480:

```
RULE WLM480: TARGET MULTIPROGRAMMING LEVEL WAS A MAJOR CAUSE OF DELAY

   The target multiprogramming level maintained by the System Resources
   Manager for Service Class BATCH (Period 1) delayed swap-in of address
   spaces in Service Class BATCH.  This finding means that an address
   space became READY, but the SRM did not start swap-in of the address
   space because of target MPL constraints.  The below information shows
   the average number of address spaces in the system, by category.
   CPExpert will produce a report at the end of this analysis which shows
   the average MPL for all service class periods.

                           AVERAGE MPL  AVG    AVG    AVG     AVG    AVG
   MEASUREMENT INTERVAL    (BATCH--1)   STC    TSO   BATCH   APPC  OPEN/MVS
   15:00-15:16,01MAR1994       8.7     97.2   51.9    13.0    0.0    0.0
```

Rule WLM480 shows the average MPL for the service class missing its performance goal, and shows the average MPL for various categories of work.

Please note that CPExpert does not produce Rule WLM480 for "served" service classes (e.g., a service class describing CICS transactions). The SRM does not collect resource information for "served" service classes. Rather, the SRM collects resource information at the "server" service class level (e.g., at the CICS region). CPExpert will analyze the "server" service class to identify constraints and Rule WLM250 may result from this analysis.

Suggestion: Rule WLM480 should never be produced for important service classes. The Workload Manager adjusts the target MPLs every 10 seconds, if necessary. The RMF measurement interval typically is at least 15 minutes. In order for Rule WLM480 to be produced, MPL delay must be a major cause of delay for the entire RMF measurement interval. This implies that higher importance work prevented swap-in of the service class period being delayed for MPL, for the entire RMF measurement interval. Such lengthy delay without Workload Manager action would be cause for considerable alarm; your system would be significantly overloaded and able to process only the higher importance work.

You may see Rule WLM480 produced often for less important service class periods. In the above example, the service class missing its performance goal consisted of batch work, and the batch work was delayed because of MPL. You may find that this delay is acceptable for such work.

When a service class fails to achieve its goal because of MPL delay, you have several alternatives:

- **Increase the importance of the service class**. The Workload Manager attempts to achieve the performance goal for each service class period.

When the Workload Manager detects that a service class period is not achieving its performance goal, the Workload Manager will assess whether changing the existing distribution of system resources will help a service class period achieve its performance goal[3].

The Workload Manager examines (and attempts to help) service class periods in descending order of importance. Importance levels may be specified as values of 1 to 5, with Importance 1 being the most important and Importance 5 being the least important. Importance 0 is an implied importance level for system tasks, and Importance 6 is an implied importance level for service class periods with a Discretionary performance goal.

If you increase the importance of a service class period, the Workload Manager will give a higher priority to the service class period when resources are allocated. Of particular relevance to the problem of a service class period being denied access to the multiprogramming set is that the Workload Manager may increase the target MPLs for the service class period if the service class period is missing its goal. With higher target MPLs, the service class period will be less likely to be delayed for MPL.

- **Decrease the importance of another service class**. The Workload Manager will attempt to provide resources to help service classes missing their performance goal. As described above, the Workload Manager examines (and attempts to help) service classes in descending order of importance.

  You should examine the importance specified for service classes with a higher importance and service classes at the same importance as the service class missing its performance goal. Determine whether these importance levels match the management objectives of your installation.

- **Alter the performance goal specified for the service class**. You should assess whether the performance goal is appropriate for the applications assigned to the service class. Perhaps the performance achieved is adequate, or perhaps the specified performance goal can be altered so that the service class meets its objective at the existing level of service. That is, the delivered service may be adequate for management objectives and you may need to change the performance goal specified to the Workload Manager.

- **Alter the performance goal specified for another service class**. You should assess whether the performance goal is appropriate for the

[3]Please refer to Section 4 for a more comprehensive discussion of the Workload Manager's algorithms.

applications assigned to other service classes. The Workload Manager attempts to achieve the performance goal for each service class. When the Workload Manager detects that a service class is not achieving its performance goal, the Workload Manager will assess whether changing the existing distribution of system resources will help a service class achieve its performance goal.

As described above, the Workload Manager first examines service classes based on importance. However, if several service classes are of the *same* importance, the Workload Manager will attempt to help the service class having the *worst* performance (as measured by the performance index).

You should assess whether appropriate performance goals have been specified for other service classes at a higher importance or at the same importance.

- **Reschedule workloads**. Your organization may be able to reschedule conflicting workloads to another system to eliminate the conflicts for processor access.

- **Improve your paging subsystem.** This option should be considered only if your system experiences significant paging delays (recall that the Workload Manager will decrease the system MPL if the paging is excessive). You can assess the paging levels by examining output from RMF or from other monitoring tools. Additionally, CPExpert will identify common problems with the paging subsystem.

- **Add another processor or acquire a faster processor**. This option should be considered only if your system is over-utilized (recall that the Workload Manager will decrease the system MPL if the system is over-utilized). You can assess the CPU utilization levels by examining output from RMF or from other monitoring tools.

- **Add additional processor storage.** This option should be considered only if your system experiences significant paging delays (recall that the Workload Manager will decrease the system MPL if the paging is excessive). You can assess the paging levels by examining output from RMF or from other monitoring tools.

- **Ignore the finding**. There may be situations in which you wish to simply ignore CPExpert's finding. You might not care that a low priority batch service class is delayed for target MPL. If this is the case, perhaps you should not have a performance goal associated with the workload.

However, you may wish to have a performance goal (and have CPExpert perform analysis) simply to assess other delays. For example, you may wish to assess the auxiliary paging delays experienced by the workload.

- **Exclude the service class from analysis**. If none of the above alternatives apply and if Rule WLM480 continually is produced for the service class, you may wish to exclude the service class from CPExpert's analysis. There is little point in having findings produced which cannot be acted upon. Please see Section 3 (Chapter 1.1.8) for information on how to exclude service classes from analysis.

After CPExpert has completed its analysis of performance constraints, a summary of MPL levels by each service class period is produced for any measurement interval in which a service class did not achieve its performance goal and the service class was delayed for MPL reasons. The following example illustrates the report which is produced:

The AVG MPL column reflects the average number of address spaces concurrently executing during the RMF measurement interval.

CPExpert annotates any service class which was delayed for target MPL as a primary or secondary cause of the service class failing to achieve its performance goal. Along with the annotation, CPExpert shows the percent of service class active time when an address space was delayed for MPL.

This report will allow you to assess the CPU time used by different service classes, by level of importance. To facilitate this review, the service class information is ordered by Importance associated with each service class.

Please note that the distribution of average MPLs may include SERVER service classes. The goal importance of the SERVER service classes is ignored after address space start-up. The importance of the SERVER service classes is a function of the service classes being served.

```
    SUMMARY OF MPL LEVELS WHEN A SERVICE CLASS WAS DELAYED FOR MPL REASONS


                        SERVICE     CLASS                 GOAL
MEASUREMENT INTERVAL     CLASS     PERIOD  GOAL TYPE     IMPORT  AVG MPL

 01MAR1994:15:00:16   SYSSTC        1    SYSTEM TASKS     0      60.0
 01MAR1994:15:00:16   SYSTEM        1    SYSTEM TASKS     0      15.0
 01MAR1994:15:00:16   MVSSUBSY      1    EX. VELOCITY     1      18.0
 01MAR1994:15:00:16   ST_USER       1    AVG RESPONSE     1       2.1
 01MAR1994:15:00:16   APPCFEED      1    EX. VELOCITY     2       1.0
 01MAR1994:15:00:16   BATCH         1    EX. VELOCITY     2       8.7  MPL DELAY(8%)
 01MAR1994:15:00:16   BERDFEED      1    EX. VELOCITY     2       0.0
 01MAR1994:15:00:16   MONITORS      1    EX. VELOCITY     2       2.0
 01MAR1994:15:00:16   ST_TOOLS      1    EX. VELOCITY     2       4.1
 01MAR1994:15:00:16   TPNSBATC      1    % RESPONSE       2       1.1
 01MAR1994:15:00:16   TPNSEVEN      1    AVG RESPONSE     2      12.8
 01MAR1994:15:00:16   TPNSEVEN      2    AVG RESPONSE     2       1.5
 01MAR1994:15:00:16   TPNSFEED      1    EX. VELOCITY     2       1.0
 01MAR1994:15:00:16   TPNSODD       1    AVG RESPONSE     2      15.1
 01MAR1994:15:00:16   TPNSODD       2    AVG RESPONSE     2       0.3
 01MAR1994:15:00:16   TPNSODD       3    AVG RESPONSE     2       0.4
 01MAR1994:15:00:16   TPNSODD       4    AVG RESPONSE     2       0.2
 01MAR1994:15:00:16   APPC          1    EX. VELOCITY     3       1.0
 01MAR1994:15:00:16   ASCH          1    EX. VELOCITY     3       2.0
 01MAR1994:15:00:16   TPNSEVEN      3    AVG RESPONSE     3       6.7
 01MAR1994:15:00:16   TPNSODD       5    AVG RESPONSE     3       1.9
 01MAR1994:15:00:16   TPNSODD       6    AVG RESPONSE     3       0.4
 01MAR1994:15:00:16   TPNSODD       7    AVG RESPONSE     3       0.1
 01MAR1994:15:00:16   VEL3          1    EX. VELOCITY     4       2.0
 01MAR1994:15:00:16   TPNSEVEN      4    DISCRETIONARY    -       2.3
 01MAR1994:15:00:16   TPNSODD       8    DISCRETIONARY    -       2.5
```

CPExpert identifies the **highest** goal importance of any served service class which had active transactions during the RMF measurement interval, and displays this highest goal importance for the server service class. **This goal importance may be different from the goal importance which was defined for the server service class using the Workload Manager ISPF panel.**

In practice, the average MPL should be relatively constant for server service classes. The address spaces associated with server service classes usually are non-swappable and typically are running for considerable periods.

Rule WLM601: XCF transport class may need to be split

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) messages were smaller than the buffer size defined for the transport class, while a significant percent of the messages were too large. Consequently, CPExpert believes that you should consider splitting the transport class.

Impact: This finding can have a LOW IMPACT or MEDIUM IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or with programs on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Within the XCF terminology, authorized programs are termed *XCF members*, and the XCF members are logically a part of specific *XCF Groups*. For example, CICS regions are considered XCF members, and the regions are logically associated with the **DFHIR000** XCF Group. RMF is logically associated with the **SYSRMF** XCF Group, the MVS Workload Manager is associated with the **SYSWLM** XCF Group, etc. One purpose of associating members with XCF groups is to facilitate system management control for similar applications.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

Different XCF groups have different signalling characteristics and different signalling performance requirements.

- For example, the Workload Manager group (SYSWLM) sends a message approximately every 10 seconds. The message is 300 bytes * the

number of service class periods with a response time or velocity goal. For a typical installation, this message might be less than 5,000 bytes. Although it is *desirable* that the Workload Manager have up-do-date information, it is not *critical* that the SYSWLM message be received at once.

- On the other hand, global resource serialization (GRS) sends such messages as the RSA-message to provide information about the serialization of global resources. The RSA-message can be sent frequently, and can be up to 32K bytes of data. It is critical to the performance of applications that the GRS message be received at once.

Optimal signalling performance requires that XCF groups have access to adequate signalling resources. These resources consist of signalling paths and buffers. Since different XCF groups have different signalling requirements, performance usually is improved if signalling resources are assigned to the XCF groups based on their requirements.

A *transport class* is the mechanism used by MVS to allow resources to be assigned to XCF groups. Resources (signalling paths, buffers, etc.) are assigned to one or more transport classes, and XCF groups are assigned to the transport classes. Thus, resources can be made available to the XCF groups as they are needed.

A particular MVS system has limited resources, and not all XCF groups require the same amount of resources. Consequently, one performance tuning consideration is the balance between (1) the resources available, (2) the resources required by different XCF groups, and (3) the value (or importance) to the installation of the various XCF group members.

The two major transport class resources to be tuned are (1) the message buffers assigned to transport classes and (2) the number of signalling paths assigned to transport classes.

The following discussion relates to the *message buffers*. Other rules in the WLM600(series) relate to the signalling paths.

Message buffers are assigned to transport classes[1] in two ways: (1) the basic assignment to the transport class via the CLASSLEN and MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

[1]Message buffers are assigned to transport classes only for **outbound** traffic since only outbound traffic can be separated into transport classes. Inbound traffic cannot be separated by transport classes; buffers are assigned to inbound traffic based on the total buffer space defined on the PATHIN statement.

- The CLASSLEN parameter defines the message length for the transport class. MVS allocates *fixed-length buffers* at the size specified in the CLASSLEN parameter for the transport class.

  If no CLASSLEN parameter is specified, MVS uses the value of the CLASSLEN parameter specified on the COUPLE statement (with a default value of 956 bytes).

  The message length specified by the CLASSLEN parameter should be large enough to accommodate most messages, but not so large as to waste storage. Selecting the correct buffer length is a tradeoff between (1) overhead incurred by having buffers too small, (2) wasted storage incurred by having buffers too large, and (3) the performance implications of mixing large and small messages in the same transport class.

  - If the fixed-length buffers are too small to hold a message, MVS acquires additional buffers to accommodate the message. Increased system overhead is caused when MVS must acquire additional buffers.

    In order to minimize this overhead, MVS may dynamically increase the length of the buffers if (1) the number of over-sized messages message traffic warrants the increase and (2) the increase in buffer length would not exceed the maximum buffer space specified on the receiving system.

  - If the buffers are too large for a message, the unused storage remaining in the buffer is wasted. This is an inefficient use of storage. Additionally, MVS could exhaust the supply of buffer space associated with a transport class if the space is wasted by specifying a buffer length that is too large for most messages. In the later case, XCF messages would be rejected if the supply of buffer space is exhausted.

  - If large and small messages are mixed in the same transport class, the small messages tend to be delayed simply because the large messages take longer to process.

- The MAXMSG parameter defines the amount of message buffer space allocated for messages sent in the transport class. The MAXMSG parameter can be specified on the PATHOUT or PATHIN statements, or on the CLASSDEF statement.

  If no MAXMSG value is specified for the paths associated with a transport class or for the CLASSDEF statement, MVS uses the value of

the MAXMSG parameter specified on the COUPLE statement (with a default of 750K bytes of buffer space).

SMF Type 74 (Subtype 2) records provide statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, how many messages were too small for the defined buffer size, how many messages fit the defined buffer size, how many messages were too big for the defined buffer size, and how many messages were over the message length for which XCF was optimized.

CPExpert analyzes this information to determine whether the correct buffer allocation has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message length specified for the transport class is inappropriate under the following conditions:

- Less than 10% of the messages **fit** the buffer length specified for the transport class. This situation, by itself, is not serious, since message lengths may be only a small amount less than the allocated buffer space.

- More than half of the messages were **smaller** than the buffer length specified for the transport class.

- A significant percent of the messages were **larger** than the buffer length specified for the transport class. This situation generates additional overhead, since MVS must either prepare additional buffer space or send additional signals to deliver the oversized messages.

  The value considered a "significant percent" of the large messages is controlled by the **PCTBIG** guidance variable. Please refer to Section 2 of this document for a discussion of the PCTBIG guidance variable.

- Additionally, CPExpert applies a "reality check" by ensuring that a reasonable number of messages were sent in the transport class.

When the above conditions are met, CPExpert produces Rule WLM601 to alert you that there is a mismatch between the buffer length specified for the transport class and the lengths of messages sent in the transport class.

The following example illustrates the output from Rule WLM601:

```
RULE WLM601: TRANSPORT CLASS MAY NEED TO BE SPLIT

   You should consider whether the DEFAULT transport class should be split.
   A large percentage of the messages were too small, while a significant
   percentage of messages were too large.  Storage is wasted when buffers
   are used by messages that are too small, while unnecessary overhead is
   incurred when XCF must expand the buffers to fit a message. The CLASSLEN
   parameter establishes the size of each message buffer, and the CLASSLEN
   parameter was specified as 16,316 for this transport class.
   This finding applies to the following RMF measurement intervals:

                           SENT    SMALL     MESSAGES   MESSAGES    TOTAL
   MEASUREMENT INTERVAL     TO    MESSAGES   THAT FIT   TOO  BIG  MESSAGES
   10:00-10:30,26MAR1996   JA0     4,296          0         57      4,353
   12:00-12:30,26MAR1996    Z0     2,653          6        762      3,421
   12:30-13:00,26MAR1996    Z0     2,017          0        109      2,126
   13:00-13:30,26MAR1996    Z0     2,543          2        180      2,743
```

Suggestion: If Rule WLM601 is regularly produced, CPExpert suggests that you consider the following alternatives[2]:

- You should evaluate the message length specified for the transport class and the message lengths of the XCF groups assigned to the transport class. You should consider "splitting" the transport class into two transport classes. Each transport class should have buffer lengths defined (using the CLASSLEN parameter) such that most of the outbound messages fit the buffer lengths defined for their respective transport classes.

  With z/OS Version 1 Release 2 (V1R2), Message IXC344I has been changed to provide more insight into the requirements of transport classes. In response to a DISPLAY XCF,CLASSDEF command, Message IXC344I displays detailed data for specific transport classes. With z/OS V1R2, the message has been enhanced to provide counts of messages sent at **each different signal size that was used**. By examining the count of messages sent at the appropriate signal size, you can determine whether the transport class should be split, and what the new sizes should be.

  If most of the outbound messages do not fit the buffer lengths, it normally is better for the buffer lengths to be slightly larger than the outbound messages. A small amount of wasted storage usually has less performance impact than the unnecessary overhead caused by messages being larger than the buffer length.

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

A major disadvantage of this approach is that signalling paths are associated with transport classes. If you "split" the transport class, you must either (1) divide the signalling paths between the new transport classes or (2) acquire additional signalling paths.

- The performance impact of having to split the signalling paths into two transport classes may outweigh the performance impact of having a mismatch between message length and buffer length.

- You may be unable to acquire additional signalling paths. However, you may find that excess signalling paths may have been assigned to **other** transport classes, and simply reassigning the signalling paths may be an acceptable alternative.

- You should evaluate whether the XCF groups are properly assigned to transport classes. XCF groups are assigned to transport classes via the GROUP parameter on the CLASSDEF statement.

  - XCF groups can be assigned to more than one transport class. When evaluating which transport class to use (when XCF groups are assigned to more than one transport class) XCF will select the transport class with the smallest buffer that will hold the message being sent. You potentially can "optimize" the buffer space used by assigning XCF groups to more than one transport class.

    All groups assigned to a transport class have equal access to the signalling resources of that class. Consequently, you should make sure that you do not assign "low priority" groups to transport classes that have high performance requirements if the "low priority" groups could cause performance degradation to the "high priority" groups.

    Fortunately, SMF Type 74 (Subtype 2) records contain information about the XCF groups and XCF members, including the number of signals sent and received by each member. This information is in the **Member Data Section** of the Type 74 records, and can be analyzed to assess the impact of message traffic of the XCF members and XCF groups.

  - Alternatively, it may be preferable to reassign XCF groups to transport classes. In practice, this situation is unlikely to occur as most installations will have a relatively small number of transport classes.

- You can adjust CPExpert's analysis by altering the value specified for the **PCTBIG** guidance variable in USOURCE(WLMGUIDE). The default value for PCTBIG is intended to cause Rule WLM601 to be produced

when more than a modest number of the messages cause MVS to incur unnecessary overhead for over-sized messages.

- If Rule WLM601 occurs frequently and there is no action you wish take, you can exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640 |

Rule WLM602: XCF message buffer length may be too small

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) messages were larger than the value specified in the CLASSLEN associated with the transport class.

Impact: This finding can have a LOW IMPACT or MEDIUM IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Within the XCF terminology, authorized programs are termed *XCF members*, and the XCF members are logically a part of specific *XCF Groups*. XCF group members communicate with each other using the XCF *signalling* mechanism.

Optimal signalling performance requires that XCF groups have access to adequate signalling resources. These resources consist of signalling paths and buffers. A *transport class* is the mechanism used by MVS to allow resources to be assigned to XCF groups. Resources (signalling paths, buffers, etc.) are assigned to one or more transport classes, and XCF groups are assigned to the transport classes. Thus, resources can be made available to the XCF groups as they are needed.

The two major transport class resources to be tuned are (1) the message buffers assigned to transport classes and (2) the number of signalling paths assigned to transport classes.

The following discussion relates to the *message buffers*. Other rules in the WLM600(series) relate to the signalling paths.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF concepts.

©Copyright 1996, Computer Management Sciences, Inc.

Message buffers are assigned to transport classes[1] in two ways: (1) the basic assignment to the transport class via the CLASSLEN and MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

- The CLASSLEN parameter defines the message length for the transport class. MVS allocates *fixed-length buffers* at the size specified in the CLASSLEN parameter for the transport class.

  If no CLASSLEN parameter is specified, MVS uses the value of the CLASSLEN parameter specified on the COUPLE statement (with a default value of 956 bytes).

  The message length specified by the CLASSLEN parameter should be large enough to accommodate most messages, but not so large as to waste storage. Selecting the correct buffer length is a tradeoff between (1) overhead incurred by having buffers too small, (2) wasted storage incurred by having buffers too large, and (3) the performance implications of mixing large and small messages in the same transport class.

  - If the fixed-length buffers are too small to hold a message, MVS acquires additional buffers to accommodate the message. Increased system overhead is caused when MVS must acquire additional buffers.

    In order to minimize this overhead, MVS may dynamically increase the length of the buffers if (1) the number of over-sized messages message traffic warrants the increase and (2) the increase in buffer length would not exceed the maximum buffer space specified on the receiving system.

  - If the buffers are too large for a message, the unused storage remaining in the buffer is wasted. This is an inefficient use of storage. Additionally, MVS could exhaust the supply of buffer space associated with a transport class if the space is wasted by specifying a buffer length that is too large for most messages. In the later case, XCF messages would be rejected if the supply of buffer space is exhausted.

  - If large and small messages are mixed in the same transport class, the small messages tend to be delayed simply because the large messages take longer to process.

[1]Message buffers are assigned to transport classes only for **outbound** traffic since only outbound traffic can be separated into transport classes. Inbound traffic cannot be separated by transport classes; buffers are assigned to inbound traffic based on the total buffer space defined on the PATHIN statement.

- The MAXMSG parameter defines the amount of message buffer space allocated for messages sent in the transport class. The MAXMSG parameter can be specified on the PATHOUT or PATHIN statements, or on the CLASSDEF statement.

  If no MAXMSG value is specified for the paths associated with a transport class or for the CLASSDEF statement, MVS uses the value of the MAXMSG parameter specified on the COUPLE statement (with a default of 750K bytes of buffer space).

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, how many messages were too small for the defined buffer size, how many messages fit the defined buffer size, how many messages were too big for the defined buffer size, and how many messages were over the message length for which XCF was optimized.

CPExpert analyzes this information to determine whether the correct buffer allocation has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message length specified for the transport class is too small when a significant percent of the messages were **larger** than the buffer length specified for the transport class and a significant percent of these messages caused overhead.

The value considered a "significant percent" of the large messages is controlled by the **PCTBIG** guidance variable. Please refer to Section 2 of this document for a discussion of the PCTBIG guidance variable.

Additionally, CPExpert applies a "reality check" by ensuring that a reasonable number of messages were sent in the transport class.

When the above conditions are met, CPExpert produces Rule WLM602 to alert you that there is a mismatch between the buffer length specified for the transport class and the lengths of messages sent in the transport class. Rule WLM602 is not produced for measurement intervals in which Rule WLM601 is produced.

The following example illustrates the output from Rule WLM602:

```
RULE WLM602: XCF MESSAGE BUFFERS MAY BE TOO SMALL

   The XCF message buffer length may be too small for the DEFAULT transport
   class. Unnecessary overhead is incurred when XCF must expand the buffers
   to fit a message.  The CLASSLEN parameter was specified as 16,316 for
   this transport class.  You should consider increasing the message length
   for this transport class or you may wish to split the transport class,
   depending upon actual message lengths.  This finding applies to the
   following RMF measurement intervals:

                           SENT    SMALL     MESSAGES   MESSAGES    TOTAL
   MEASUREMENT INTERVAL     TO    MESSAGES   THAT FIT   TOO  BIG  MESSAGES
   10:00-10:30,26MAR1996   J80     2,654          0        462      3,116
   11:30-12:00,26MAR1996   J80     3,006          1        381      3,388
   12:00-12:30,26MAR1996   J80     3,481          0        493      3,884
   12:00-12:30,26MAR1996    Z0     2,943          6        472      3,421
```

Suggestion: If Rule WLM602 is regularly produced, CPExpert suggests that you consider the following alternatives[2]:

- You should evaluate the message length specified for the transport class and the message lengths of the XCF groups assigned to the transport class. You should consider using the CLASSLEN parameter of the CLASSDEF statement to increase the message length for the transport class.

  With z/OS Version 1 Release 2 (V1R2), Message IXC344I has been changed to provide more insight into the requirements of transport classes. In response to a DISPLAY XCF,CLASSDEF command, Message IXC344I displays detailed data for specific transport classes. With z/OS V1R2, the message has been enhanced to provide counts of messages sent at **each different signal size that was used**. By examining the count of messages sent at the appropriate signal size, you can determine whether the transport class should be split, and what the new sizes should be.

  If most of the outbound messages do not fit the buffer lengths, it normally is better for the buffer lengths to be slightly larger than the outbound messages. A small amount of wasted storage usually has less performance impact than the unnecessary overhead caused by messages being larger than the buffer length.

- You should evaluate whether the XCF groups are properly assigned to transport classes. XCF groups are assigned to transport classes via the GROUP parameter on the CLASSDEF statement.

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

- XCF groups can be assigned to more than one transport class. When evaluating which transport class to use (when XCF groups are assigned to more than one transport class) XCF will select the transport class with the smallest buffer that will hold the message being sent. You potentially can "optimize" the buffer space used by assigning XCF groups to more than one transport class.

  All groups assigned to a transport class have equal access to the signalling resources of that class. Consequently, you should make sure that you do not assign "low priority" groups to transport classes that have high performance requirements if the "low priority" groups could cause performance degradation to the "high priority" groups.

  Fortunately, SMF Type 74 (Subtype 2) records contain information about the XCF groups and XCF members, including the number of signals sent and received by each member. This information is in the **Member Data Section** of the Type 74 records, and can be analyzed to assess the impact of message traffic of the XCF members and XCF groups.

- Alternatively, it may be preferable to reassign XCF groups to transport classes. In practice, this situation is unlikely to occur as most installations will have a relatively small number of transport classes.

- You can adjust CPExpert's analysis by altering the value specified for the **PCTBIG** guidance variable in USOURCE(WLMGUIDE). The default value for PCTBIG is intended to cause Rule WLM602 to be produced when more than a modest number of the messages cause MVS to incur unnecessary overhead for over-sized messages.

- If Rule WLM602 occurs frequently and there is no action you wish take, you can exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640 |

Rule WLM603: XCF message buffer length may be too large

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) messages were smaller than the value specified in the CLASSLEN associated with the transport class.

Impact: This finding can have a LOW IMPACT on the signalling performance of the sysplex. The finding can have a more significant impact on performance of the overall system, since central storage may be wasted by the allocation of unused storage.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Within the XCF terminology, authorized programs are termed *XCF members*, and the XCF members are logically a part of specific *XCF Groups*. XCF group members communicate with each other using the XCF *signalling* mechanism.

Optimal signalling performance requires that XCF groups have access to adequate signalling resources. These resources consist of signalling paths and buffers. A *transport class* is the mechanism used by MVS to allow resources to be assigned to XCF groups. Resources (signalling paths, buffers, etc.) are assigned to one or more transport classes, and XCF groups are assigned to the transport classes. Thus, resources can be made available to the XCF groups as they are needed.

The two major transport class resources to be tuned are (1) the message buffers assigned to transport classes and (2) the number of signalling paths assigned to transport classes.

The following discussion relates to the *message buffers*. Other rules in the WLM600(series) relate to the signalling paths.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF concepts.

©Copyright 1996, Computer Management Sciences, Inc.

Message buffers are assigned to transport classes[1] in two ways: (1) the basic assignment to the transport class via the CLASSLEN and MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

- The CLASSLEN parameter defines the message length for the transport class. MVS allocates *fixed-length buffers* at the size specified in the CLASSLEN parameter for the transport class.

  If no CLASSLEN parameter is specified, MVS uses the value of the CLASSLEN parameter specified on the COUPLE statement (with a default value of 956 bytes).

  The message length specified by the CLASSLEN parameter should be large enough to accommodate most messages, but not so large as to waste storage. Selecting the correct buffer length is a tradeoff between (1) overhead incurred by having buffers too small, (2) wasted storage incurred by having buffers too large, and (3) the performance implications of mixing large and small messages in the same transport class.

  - If the fixed-length buffers are too small to hold a message, MVS acquires additional buffers to accommodate the message. Increased system overhead is caused when MVS must acquire additional buffers.

    In order to minimize this overhead, MVS may dynamically increase the length of the buffers if (1) the number of over-sized messages message traffic warrants the increase and (2) the increase in buffer length would not exceed the maximum buffer space specified on the receiving system.

  - If the buffers are too large for a message, the unused storage remaining in the buffer is wasted. This is an inefficient use of storage. Additionally, MVS could exhaust the supply of buffer space associated with a transport class if the space is wasted by specifying a buffer length that is too large for most messages. In the later case, XCF messages would be rejected if the supply of buffer space is exhausted.

  - If large and small messages are mixed in the same transport class, the small messages tend to be delayed simply because the large messages take longer to process.

[1]Message buffers are assigned to transport classes only for **outbound** traffic since only outbound traffic can be separated into transport classes. Inbound traffic cannot be separated by transport classes; buffers are assigned to inbound traffic based on the total buffer space defined on the PATHIN statement.

- The MAXMSG parameter defines the amount of message buffer space allocated for messages sent in the transport class. The MAXMSG parameter can be specified on the PATHOUT or PATHIN statements, or on the CLASSDEF statement.

  If no MAXMSG value is specified for the paths associated with a transport class or for the CLASSDEF statement, MVS uses the value of the MAXMSG parameter specified on the COUPLE statement (with a default of 750K bytes of buffer space).

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, how many messages were too small for the defined buffer size, how many messages fit the defined buffer size, how many messages were too big for the defined buffer size, and how many messages were over the message length for which XCF was optimized.

CPExpert analyzes this information to determine whether the correct buffer allocation has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message length specified for the transport class is too large when a significant percent of the messages were **smaller** than the buffer length specified for the transport class.

The value considered a "significant percent" of the messages is controlled by the **PCTSML** guidance variable. Please refer to Section 2 of this document for a discussion of the PCTSML guidance variable.

Additionally, CPExpert applies a "reality check" by ensuring that a reasonable number of messages were sent in the transport class.

When the above conditions are met, CPExpert produces Rule WLM603 to alert you that there is a mismatch between the buffer length specified for the transport class and the lengths of messages sent in the transport class. Rule WLM603 is not produced for measurement intervals in which Rule WLM601 is produced.

The following example illustrates the output from Rule WLM603:

```
RULE WLM603: XCF MESSAGE BUFFER LENGTH MAY BE TOO LARGE

   The XCF message buffer length may be too large for the DEFAULT transport
   class.  XCF will fill the message buffer space too quickly when the
   specified message length is larger than most of the messages sent.
   The CLASSLEN parameter was specified as 16,316 for this transport class,
   and over 99% of the messages were less than this length.  You should
   consider decreasing the message length for this transport class or you
   may wish to split the transport class, depending upon actual message
   lengths.  This situation is not critical, since XCF did not exhaust its
   message buffer space.  The finding is produced only to alert you to a
   potential problem with storage allocation.  You can suppress this finding
   by altering the PCTSML guidance variable in USOURCE(WLMGUIDE).  This
   finding applies to the following RMF measurement intervals:

                           SENT    SMALL     MESSAGES   MESSAGES    TOTAL
   MEASUREMENT INTERVAL     TO    MESSAGES   THAT FIT   TOO  BIG  MESSAGES
   13:00-13:30,26MAR1996   J90     2,159          1          0      2,160
   13:00-13:30,26MAR1996   JB0     2,263          0          0      2,263
```

Suggestion: If Rule WLM603 is regularly produced, CPExpert suggests that you consider the following alternatives[2]:

- You should evaluate the message length specified for the transport class and the message lengths of the XCF groups assigned to the transport class. You should consider using the CLASSLEN parameter of the CLASSDEF statement to decrease the message length for the transport class.

  With z/OS Version 1 Release 2 (V1R2), Message IXC344I has been changed to provide more insight into the requirements of transport classes. In response to a DISPLAY XCF,CLASSDEF command, Message IXC344I displays detailed data for specific transport classes. With z/OS V1R2, the message has been enhanced to provide counts of messages sent at **each different signal size that was used**. By examining the count of messages sent at the appropriate signal size, you can determine whether the transport class should be split, and what the new sizes should be.

  If most of the outbound messages do not fit the buffer lengths, it normally is better for the buffer lengths to be slightly larger than the outbound messages. A small amount of wasted storage usually has less performance impact than the unnecessary overhead caused by messages being larger than the buffer length.

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

- You should evaluate whether the XCF groups are properly assigned to transport classes. XCF groups are assigned to transport classes via the GROUP parameter on the CLASSDEF statement.

  - XCF groups can be assigned to more than one transport class. When evaluating which transport class to use (when XCF groups are assigned to more than one transport class) XCF will select the transport class with the smallest buffer that will hold the message being sent. You potentially can "optimize" the buffer space used by assigning XCF groups to more than one transport class.

    All groups assigned to a transport class have equal access to the signalling resources of that class. Consequently, you should make sure that you do not assign "low priority" groups to transport classes that have high performance requirements if the "low priority" groups could cause performance degradation to the "high priority" groups.

    Fortunately, SMF Type 74 (Subtype 2) records contain information about the XCF groups and XCF members, including the number of signals sent and received by each member. This information is in the **Member Data Section** of the Type 74 records, and can be analyzed to assess the impact of message traffic of the XCF members and XCF groups.

  - Alternatively, it may be preferable to reassign XCF groups to transport classes. In practice, this situation is unlikely to occur as most installations will have a relatively small number of transport classes.

- You can adjust CPExpert's analysis by altering the value specified for the **PCTSML** guidance variable in USOURCE(WLMGUIDE). The default value for PCTSML is intended to cause Rule WLM603 to be produced when more than 90% of the messages are smaller than the defined buffer length. You can alter the analysis by specifying a different value (and you can override the analysis completely by specifying **%LET PCTSML = 100;** for the guidance).

- If Rule WLM603 occurs frequently and there is no action you wish take, you can exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

Rule WLM604: XCF outbound message buffer space may be too small

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) outbound messages were rejected because of constraints on the amount of outbound message buffer space.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

Message buffer space for outbound traffic is assigned to transport classes[1] in two ways: (1) the basic assignment to the transport class via the MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the PATHOUT statement.

The MAXMSG parameter defines the amount of message buffer space allocated for outbound messages sent in the transport class. The MAXMSG parameter can be specified on the PATHOUT statements, or on the CLASSDEF statement. The message buffer space available to outbound messages in a transport class is the sum of the message buffer space specified for the transport class on the CLASSDEF statement, plus the message buffer space specified for each outbound signalling path assigned to the transport class.

If no MAXMSG value is specified for the paths associated with a transport class or for the CLASSDEF statement, MVS uses the value of the MAXMSG parameter specified on the COUPLE statement (with a default of 750K bytes of buffer space).

[1]Message buffers are assigned to transport classes only for **outbound** traffic since only outbound traffic can be separated into transport classes. Inbound traffic cannot be separated by transport classes; buffers are assigned to inbound traffic based on the total buffer space defined on the PATHIN statement.

©Copyright 1996, Computer Management Sciences, Inc.

Message buffer space for **outbound** messages is separated by transport class, so a sudden high volume of traffic in one transport class will not cause performance problems for another transport class. If the message buffer space required to support messages in a particular transport class is exhausted, MVS will reject additional messages until outbound message buffer space becomes available in the transport class.

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, how many messages were rejected because there was insufficient message buffer space, and how much message buffer space was allocated to the transport class.

CPExpert analyzes this information to determine whether sufficient message buffer space has been defined. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that the message buffer space is too small for the transport class when more than the value specified for the **PCTREJ** guidance variable of the outbound messages were rejected because of no buffer space. The default specification for the PCTREJ guidance variable is **%LET PCTREJ = 0.1;** indicating that Rule WLM604 will be produced when more than one-tenth of a percent of the outbound traffic is rejected for insufficient buffer space.

CPExpert produces Rule WLM604 to alert you that a significant percent of messages have been rejected because of insufficient buffer space.

The following example illustrates the output from Rule WLM604:

```
RULE WLM604: THE XCF MESSAGE BUFFER SPACE MAY BE TOO SMALL

   The message buffer space may be too small for the DEFAULT transport
   class.  CPExpert noticed that XCF requests were rejected because of
   constraints on the amount of message buffer space.  You should consider
   increasing the amount of XCF message buffer space for the DEFAULT
   transport class.  An asterisk beside the buffer space means that the
   buffer space DECREASED during the reported measurement interval, from
   the preceding measurement interval.  This finding applies to the
   following measurement intervals:
```

MEASUREMENT INTERVAL	SENT TO	TOTAL REQUESTS	REJECTED REQUESTS	PCT REJECTED	BUFFER SPACE
13:00-13:30,26MAR1996	J90	2,160	196	9.1	1,536K
13:00-13:30,26MAR1996	JA0	587	49	8.3	1,536K
13:00-13:30,26MAR1996	JB0	2,263	174	7.7	1,536K
13:00-13:30,26MAR1996	JC0	1,492	107	7.1	1,536K
13:00-13:30,26MAR1996	Z0	1,086	60	5.6	1,536K
13:00-13:30,26MAR1996	Z1	203	11	5.3	1,536K

Suggestion: The available outbound buffer space for a transport class can be too small because (1) the amount initially specified was too low, (2) a system operator could have decreased the amount of message buffer space for the transport class, or (3) there could have been a loss of one or more paths assigned to the transport class.

If Rule WLM604 is produced, CPExpert suggests that you consider the following alternatives[2]:

- You should assess whether a system operator changed the amount of message buffer space assigned to the transport class or to paths assigned to the transport class. CPExpert will notify you (by placing '***' beside the buffer space value) if the amount of allocated message buffer space assigned to the transport class **decreased** from the previous RMF measurement interval.

  If the system operator did make a change resulting in less outbound message buffer space for the transport class, you should verify that there was a sound rationale for the action.

- You should evaluate the amount of message space specified for the transport class on the CLASSDEF statement and the amount of message buffer space specified for each path assigned to the transport class. You should consider using the MAXMSG parameter of the CLASSDEF statement or the PATHOUT statement to increase the message buffer space for the transport class.

- You should assess whether there has been a decrease in the number of paths assigned to the transport class. Since the available message buffer space for transport classes is partly a function of the message buffer space assigned to paths associated with the transport class (for output messages), a decrease in the number of paths would cause a decrease in the message buffer space. A system operator could have issued the SETXCF STOP command to delete a signalling path, or a path could have failed.

- If Rule WLM604 occurs frequently and there is no action you wish take, you should change the guidance to CPExpert by altering the PCTREJ guidance variable in USOURCE(WLMGUIDE).

  Alternatively, you can use the **EXCLASSn** guidance variables to exclude the transport class from CPExpert's analysis. The EXCLASSn guidance

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters) Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640 |

Rule WLM605: XCF inbound message buffer space may be too small

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) inbound messages were rejected because of constraints on the amount of inbound message buffer space.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

Inbound message buffers are used to receive messages from another system. These buffers are allocated, as needed, to support the message traffic load. Message buffer space for **inbound** messages is separated by signalling path.

Message buffer space for inbound traffic is assigned by the MAXMSG parameter on the PATHIN statement for each inbound signalling path. If no MAXMSG parameter is specified, the value on the MAXMSG parameter of the COUPLE statement is used as a default buffer space specification.

Message buffers associated with an inbound signalling path do not receive messages over any other inbound signalling path. If the inbound message buffer space required to support messages on a particular inbound signalling path is exhausted, MVS will reject additional messages until message buffer space becomes available in for the inbound signalling path.

SMF Type 74 (Subtype 2) provides statistics about the number of inbound messages received, where the messages are sent, how many messages were rejected because there was insufficient message buffer space, and how much input message buffer space was allocated.

©Copyright 1996, Computer Management Sciences, Inc.

CPExpert analyzes this information to determine whether sufficient message buffer space has been defined. CPExpert computes the total inbound message traffic. CPExpert concludes that the inbound message buffer space is too small when more than the value specified for the **PCTREJ** guidance variable of the inbound messages were rejected because of no buffer space. The default specification for the PCTREJ guidance variable is **%LET PCTREJ = 0.1;** indicating that Rule WLM605 will be produced when more than one-tenth of a percent of the inbound traffic is rejected for insufficient buffer space.

CPExpert produces Rule WLM605 to alert you that a significant percent of inbound messages have been rejected because of insufficient buffer space.

The following example illustrates the output from Rule WLM605:

```
RULE WLM605: THE XCF INBOUND MESSAGE BUFFER SPACE MAY BE TOO SMALL

   The inbound message buffer space may be too small.  CPExpert noticed
   that XCF input requests were rejected because of constraints on the
   amount of input message buffer space.  An asterisk beside the buffer
   space means that the buffer space DECREASED during the reported
   measurement interval, from the preceding measurement interval.  You
   should consider increasing the amount of input message buffer space.
   This finding applies to the following measurement intervals:

                          RECEIVED   TOTAL    REJECTED      PCT     BUFFER
   MEASUREMENT INTERVAL     FROM   REQUESTS   REQUESTS   REJECTED    SPACE
   13:00-13:30,26MAR1996    J80      9,242       462        5.0       500K ***
```

Suggestion: The available inbound buffer space for an inbound path can be too small because (1) the amount initially specified on the PATHIN statement was too low, (2) a system operator could have decreased the amount of inbound message buffer space for one or more paths, or (3) one or more paths have been deleted or have failed.

If Rule WLM605 is produced, CPExpert suggests that you consider the following alternatives[1]:

- You should evaluate the amount of message space specified on the MAXMSG parameter of the PATHIN statement. You should consider increasing the inbound message buffer space.

[1]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

- You should assess whether a system operator changed the amount of inbound message buffer space assigned to an inbound path. CPExpert will notify you (by placing '***' beside the buffer space value) if the amount of allocated message buffer space assigned to the inbound path **decreased** from the previous RMF measurement interval.

  If the system operator did make a change resulting in less message buffer space for an inbound path, you should verify that there was a sound rationale for the action.

- You should assess whether there has been a decrease in the number of inbound paths. A system operator could have issued the SETXCF STOP command to delete a signalling path, or a path could have failed.

- When Rule WLM605 is produced, CPExpert often will produce Rule WLM620 to identify the outbound/inbound path combination that is experiencing problems.

  - It is possible that Rule WLM605 would be produced but CPExpert cannot identify an outbound/inbound path combination causing problems. This situation could occur when there is a **general** problem with the inbound buffer space over all paths, but no path combination causes the problem.

  - It is possible that Rule WLM605 would **not** be produced, but CPExpert could produce Rule WLM620. This situation could occur when there is not a **general** problem with the inbound buffer space for all paths, but a particular outbound/inbound path combination is experiencing problems.

    Rule WLM605 is based on the PCTREJ guidance variable, which guides the assessment of rejects of outbound messages (analyzing SMF Type 74, Subtype 2, System Data). Rule WLM620 is based on comparing the outbound path BUSY with the inbound path BUFFER UNAVAILABLE condition (analyzing SMF Type 74, Subtype 2, Path Data).

    Since different data are analyzed by different logic paths, it is not always possible for CPExpert to produce both Rule WLM605 and Rule WLM620.

- If Rule WLM605 occurs frequently and there is no action you wish take, you should change the guidance to CPExpert by altering the PCTREJ guidance variable in USOURCE(WLMGUIDE).

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640 |

©Copyright 1996, Computer Management Sciences, Inc.

Rule WLM606: XCF local message buffer space may be too small

Finding: CPExpert has determined that a large percent of the cross system coupling facility (XCF) local messages were rejected because of constraints on the amount of local message buffer space.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

Local message buffers are used to send and receive messages from programs within the same system. These buffers are allocated, as needed, to support the message traffic load.

Message buffer space for **local** messages is separated by transport class, so a sudden high volume of traffic in one transport class will not cause performance problems for another transport class. If the message buffer space required to support messages in a particular transport class is exhausted, MVS will reject additional messages until local message buffer space becomes available in the transport class.

Message buffer space for local traffic is assigned to transport classes in two ways: (1) the basic assignment to the transport class via the MAXMSG parameters on the CLASSDEF statement and (2) the MAXMSG parameter on the LOCALMSG statement. If no MAXMSG value is specified for the CLASSDEF statement, MVS uses the value of the MAXMSG parameter specified on the COUPLE statement (with a default of 750K bytes of buffer space). If the LOCALMSG statement is provided, the MAXMSG parameter must be specified.

The local message buffer space in a transport class is the **sum** of the message buffer space specified for the transport class on the CLASSDEF

statement, plus the message buffer space specified on the LOCALMSG statement.

SMF Type 74 (Subtype 2) provides statistics about the number of local messages sent by XCF groups in a transport class, how many messages were rejected because there was insufficient local message buffer space, and how much local message buffer space was allocated to the transport class.

CPExpert analyzes this information to determine whether sufficient local message buffer space has been defined. CPExpert computes the total local message traffic for a transport class. CPExpert concludes that the message buffer space is too small for the transport class when more than the value specified for the **PCTREJ** guidance variable of the local messages were rejected because of no buffer space. The default specification for the PCTREJ guidance variable is **%LET PCTREJ = 0.1;** indicating that Rule WLM606 will be produced when more than one-tenth of a percent of the local traffic is rejected for insufficient buffer space.

CPExpert produces Rule WLM606 to alert you that a significant percent of messages have been rejected because of insufficient buffer space.

The following example illustrates the output from Rule WLM606:

```
RULE WLM606: THE XCF LOCAL MESSAGE BUFFER SPACE MAY BE TOO SMALL

   The local message buffer space may be too small.  CPExpert noticed that
   XCF local requests were rejected because of constraints on the amount
   of message buffer space.  An asterisk beside the buffer space means
   that the local buffer space DECREASED during the reported measurement
   interval, from the preceeding measurement interval.  You should consider
   increasing the amount of local message buffer space.  This finding
   applies to the following measurement intervals:

                                  TOTAL    REJECTED      PCT       BUFFER
   MEASUREMENT INTERVAL         REQUESTS   REQUESTS   REJECTED      SPACE
   10:00-10:30,26MAR1996          4,406        68        1.5         942K ***
   10:30-11:00,26MAR1996          4,821        73        1.5         942K
   11:00-11:30,26MAR1996          4,456        67        1.5         942K
   11:30-12:00,26MAR1996          3,991        59        1.5         942K
```

Suggestion: The available local buffer space for a transport class can be too small because (1) the amount initially specified was too low or (2) a system operator could have decreased the amount of message buffer space for the transport class.

If Rule WLM606 is produced, CPExpert suggests that you consider the following alternatives[1]:

- You should assess whether a system operator changed the amount of local message buffer space assigned to the transport class. CPExpert will notify you (by placing '***' beside the buffer space value) if the amount of local message buffer space assigned to the transport class **decreased** from the previous RMF measurement interval.

  If the system operator did make a change resulting in less local message buffer space for the transport class, you should verify that there was a sound rationale for the action.

- You should evaluate the amount of local message buffer space specified for the transport class on the LOCALMSG statement, CLASSDEF statement, or the COUPLE statement. You should consider using the MAXMSG parameter of the LOCALMSG statement to increase the message buffer space for the transport class.

- If Rule WLM606 occurs frequently and there is no action you wish take, you should change the guidance to CPExpert by altering the PCTREJ guidance variable in USOURCE(WLMGUIDE).

  Alternatively, you can use the **EXCLASSn** guidance variables to exclude the transport class from CPExpert's analysis. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

[1]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

z/OS V1R2: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R3: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

z/OS V1R4: MVS System Messages, Volume 10 (IXP-IZP), SA22-7640

Rule WLM607: Insufficient outbound paths were defined

Finding: There was a significant value in the "ALL PATHS UNAVAILABLE" field for a transport class. Any significant value in the "ALL PATHS UNAVAILABLE" field for a transport class usually indicates that you have too few outbound paths defined. Alternatively, there may be an error in the path definitions (for example, you may have a typographical error).

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon how often the "ALL PATHS UNAVAILABLE" condition was experienced, and on the message characteristics of (1) the transport class experiencing "ALL PATHS UNAVAILABLE" and (2) the transport classes to which XCF routes messages.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

Outbound paths are assigned to transport classes by using the CLASS parameter on the PATHOUT statement (or by using the SETXCF PATH command after IPL). At least one outbound signalling path should be assigned to each transport class[1]. If there is high message traffic in the transport class, you may wish to assign **more** than one signalling path to

[1]You are not required to assign a signalling path to a transport class. If no signalling path is assigned to a transport class, the XCF groups in the transport class compete for signalling resources of other transport classes. This situation can degrade signalling performance.

©Copyright 1996, Computer Management Sciences, Inc.

the transport class. Additionally, you may wish to assign more signalling paths for redundancy.

XCF usually uses sends messages only on the signalling paths associated with a transport class. However, if there are no paths available to the transport class, XCF will route messages to other paths. Routing messages to other paths generates additional overhead for XCF to send the outbound message.

Additionally, using other paths may cause conflicts with the normal message traffic on these paths. As described in Rule WLM601, some messages are long and some are short; some messages have critical timing for system performance and some are less critical. If non-critical messages are routed to paths associated with a transport class with critical traffic, any resulting delays to the critical traffic could cause overall system performance problems.

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, and how often the "ALL PATHS UNAVAILABLE" condition was experienced.

CPExpert analyzes this information to determine whether the XCF experienced the "ALL PATHS UNAVAILABLE" condition for an excessive percent of the outbound messages. CPExpert computes the total outbound message traffic for a transport class. CPExpert concludes that too few paths may be available to the transport class when the "ALL PATHS UNAVAILABLE" condition occurred more than one percent of the outbound messages for the transport class.

CPExpert produces Rule WLM607 to alert you that there may be too few outbound paths assigned to the transport class.

The following example illustrates the output from Rule WLM607:

```
RULE WLM607: THERE MAY BE AN ERROR IN PATH DEFINITION

   There was a significant value in the "ALL PATHS UNAVAILABLE" field for
   the DEFSMALL transport class.  Any significant value in the "ALL PATHS
   UNAVAILABLE" field usually indicates that there may be an error in the
   path definitions (for example, you may have a typographical error).
   This finding applies to the following RMF measurement intervals:

                                                        PCT ALL PATHS   NUMBER
   MEASUREMENT INTERVAL        MESSAGES     SENT TO     UNAVAILABLE   OF PATHS
   13:00-13:30,26MAR1996        63,359       J90            4.8           2
   13:00-13:30,26MAR1996        34,633       JB0            4.3           2
   13:00-13:30,26MAR1996        28,471       JC0            4.2           2
   13:00-13:30,26MAR1996        26,648       JD0            4.0           2
   13:00-13:30,26MAR1996        21,621       JE0            3.8           2
   13:00-13:30,26MAR1996        20,652       JF0            3.7           2
```

Suggestion: If Rule WLM607 is regularly produced, CPExpert suggests that you consider the following alternatives[2]:

- A likely cause of the "ALL PATHS UNAVAILABLE" condition is that you may have an error in the path definitions (for example, you may have a typographical error) in the DEVICE parameter list or the STRNAME parameter list of the PATHOUT statement associated with the transport class. Please verify that the list is correct.

- You should determine whether the number of outbound paths for the transport class is less than you defined. The number of paths can decease because a path failed or because an operator deleted a path (using the SETXCF STOP, PATHOUT,DEVICE=outdevnum) command. CPExpert will display the number of paths for any RMF interval in which Rule WLM607 is produced. Please compare the number of paths with the number of paths specified on the DEVICE parameter or STRNAME parameter of the PATHOUT statement associated with the transport class.

- You should evaluate the number of outbound paths specified for the transport class. You should examine the DEVICE parameter or the STRNAME parameter list of the PATHOUT statement associated with the transport class to determine whether additional paths should be assigned.

  In evaluating the number of paths assigned to the transport class, you should consider (1) the importance of the messages in the transport class, (2) how often the "ALL PATHS UNAVAILABLE" condition was experienced, and (3) the potential impact on other transport classes when XCF must route the outbound messages to paths assigned to other transport classes.

- You should evaluate whether the XCF groups are properly assigned to transport classes. It may be preferable to reassign XCF groups to transport classes. In practice, this situation is unlikely to occur as most installations will have a relatively small number of transport classes.

  - XCF groups can be assigned to more than one transport class. When evaluating which transport class to use (when XCF groups are assigned to more than one transport class) XCF will select the transport class with the smallest buffer that will hold the message

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

being sent. You potentially can "optimize" the buffer space used by assigning XCF groups to more than one transport class.

All groups assigned to a transport class have equal access to the signalling resources of that class. Consequently, you should make sure that you do not assign "low priority" groups to transport classes that have high performance requirements if the "low priority" groups could cause performance degradation to the "high priority" groups.

Fortunately, SMF Type 74 (Subtype 2) records contain information about the XCF groups and XCF members, including the number of signals sent and received by each member. This information is in the **Member Data Section** of the Type 74 records, and can be analyzed to assess the impact of message traffic of the XCF members and XCF groups.

- If Rule WLM607 occurs frequently and there is no action you wish take, you can to exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM608: Transport class did not have a signalling path assigned

Finding: The transport class did not have a signalling path assigned. There may be an error in the path definitions (for example, you may have a typographical error). Alternatively, the path(s) assigned to the transport class might have failed or have been deleted by an operator.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon (1) the number of messages sent in the transport class, (2) the message characteristics of the transport class having no path assigned, and (3) the transport classes to which XCF routes messages.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

Outbound paths are assigned to transport classes by using the CLASS parameter on the PATHOUT statement (or by using the SETXCF PATH command after IPL). At least one outbound signalling path should be assigned to each transport class[1]. If there is high message traffic in the transport class, you may wish to assign **more** than one signalling path to the transport class. Additionally, you may wish to assign more signalling paths for redundancy.

[1]You are not required to assign a signalling path to a transport class. If no signalling path is assigned to a transport class, the XCF groups in the transport class compete for signalling resources of other transport classes. This situation can degrade signalling performance.

©Copyright 1996, Computer Management Sciences, Inc.

XCF usually uses sends messages only on the signalling paths associated with a transport class. However, if there are no paths available to the transport class, XCF will route messages to other paths. Routing messages to other paths generates additional overhead for XCF to send the outbound message.

Additionally, using other paths may cause conflicts with the normal message traffic on these paths. As described in Rule WLM601, some messages are long and some are short; some messages have critical timing for system performance and some are less critical. If non-critical messages are routed to paths associated with a transport class with critical traffic, any resulting delays to the critical traffic could cause overall system performance problems.

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, and how many paths were assigned to the transport class.

CPExpert analyzes this information to determine whether at least one path was assigned to the transport class. CPExpert produces Rule WLM608 when there were no paths assigned to a transport class. Before firing Rule WLM608, CPExpert applies a "reality check" to make sure that a reasonable amount of traffic was sent in the transport class.

The following example illustrates the output from Rule WLM608:

```
RULE WLM608: TRANSPORT CLASS DID NOT HAVE SIGNALLING PATH ASSIGNED

   The DEFAULT Transport Class did not have a signalling path assigned,
   yet there was activity on the transport class.  Performance is degraded
   when a transport class does not have a signalling path assigned, since
   the groups compete for the signalling resources of transport classes
   assigned to other XCF groups.  This finding applies to the following RMF
   measurement intervals:

                                                TOTAL        MESSAGE
   MEASUREMENT INTERVAL        SENT TO       REQUESTS        LENGTH
   13:00-13:30,26MAR1996         J90           2,160        16,316
   13:00-13:30,26MAR1996         JA0             587        16,316
   13:00-13:30,26MAR1996         JB0           2,263        16,316
   13:00-13:30,26MAR1996         JC0           1,492        16,316
   13:00-13:30,26MAR1996         JD0           1,336        16,316
   13:00-13:30,26MAR1996         JE0             898        16,316
   13:00-13:30,26MAR1996         JF0             840        16,316
   13:00-13:30,26MAR1996          Z0           1,086        16,316
   13:00-13:30,26MAR1996          Z1             203        16,316
```

Suggestion: If Rule WLM608 is produced, CPExpert suggests that you consider the following alternatives[2]:

- A likely cause of no paths being assigned to a transport class is that you may have an error in the path definitions (for example, you may have a typographical error) in the DEVICE parameter list of the PATHOUT statement associated with the transport class. Please verify that the list is correct.

- You should determine whether an assigned path failed, or whether an operator deleted a path (using the SETXCF STOP, PATHOUT,DEVICE=outdevnum) command.

- If no outbound paths were assigned to the transport class, you normally should assign at least one path.

- If Rule WLM608 occurs frequently and there is no action you wish take, you can to exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

---

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

Rule WLM620: Message buffer space may be too small for inbound path

Finding: CPExpert believes that the buffer space may be too small for an inbound path.

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon how often XCF was unable to obtain buffer space for inbound messages.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

Inbound message buffers are used to receive messages from another system. These buffers are allocated, as needed, to support the message traffic load. Message buffer space for **inbound** messages is separated by signalling path.

Message buffer space for inbound traffic is assigned by the MAXMSG parameter on the PATHIN statement for each inbound signalling path. If no MAXMSG parameter is specified, the value on the MAXMSG parameter of the COUPLE statement is used as a default buffer space specification.

Message buffers associated with an inbound signalling path do not receive messages over any other inbound signalling path. If the inbound message buffer space required to support messages on a particular inbound signalling path is exhausted, MVS will reject additional messages until message buffer space becomes available in for the inbound signalling path.

SMF Type 74 (Subtype 2) provides statistics about the number of inbound messages received, where the messages are sent, how many messages were rejected because there was insufficient message buffer space, and how much input message buffer space was allocated. CPExpert analyzes this information to determine whether sufficient message buffer space has been defined.

©Copyright 1996, Computer Management Sciences, Inc.

CPExpert computes the total inbound message traffic. CPExpert concludes that the inbound message buffer space is too small when more than the value specified for the **PCTREJ** guidance variable of the inbound messages were rejected because of no buffer space. The default specification for the PCTREJ guidance variable is **%LET PCTREJ = 0.1;** indicating that Rule WLM605 will be produced when more than one-tenth of a percent of the inbound traffic is rejected for insufficient buffer space.

CPExpert produces Rule WLM605 to alert you that a significant percent of inbound messages have been rejected because of insufficient buffer space. Please refer to Rule WLM605 for additional information.

As mentioned above, message buffer space for inbound messages is separated by signalling path. If the inbound signalling path on the reeiving system does not have enough buffer space, signals on the outbound path of the sending system can back up. When the outbound signals back up, the path will reflect more BUSY time.

CPExpert examines each outbound path for a high BUSY condition, and examines the corresponding inbound path on the receiving system for a high BUFFER UNAVAILABLE condition. When these conditions are met, CPExpert produces Rule WLM620 to identify the path combination that is likely causing system performance problems because of too little inbound buffer space.

The following example illustrates the output from Rule WLM620:

```
RULE WLM620: MESSAGE BUFFER SPACE MAY BE TOO SMALL FOR INBOUND PATH

   The OUTBOUND path busy was high for the C584 device on System Z1 to
   the C410 device on System J80, while the INBOUND path on System J80
   had a high BUFFER UNAVAILABLE condition.  This situation usually means
   that you should increase the message buffer space for the inbound path.
   The message buffer space currently is specified as 500K for the
   inbound path on System J80.  This finding applies to the following
   RMF measurement intervals:

                            OUTBOUND  PCT OUTBOUND  INBOUND   PCT  BUFFERS
   MEASUREMENT INTERVAL     REQUESTS  BUFFERS BUSY  REQUESTS   UNAVAILABLE
   13:00-13:30,26MAR1996       2,088       10.1        2,088        8.0
```

Suggestion: The available buffer space for an inbound path can be too small because (1) the amount initially specified on the PATHIN statement was too low or (2) a system operator could have decreased the amount of inbound message buffer space for the paths.

If Rule WLM620 is produced, CPExpert suggests that you consider the following alternatives[1]:

- You should evaluate the amount of message space specified on the MAXMSG parameter of the PATHIN statement. You should consider increasing the inbound message buffer space.

- You should assess whether a system operator changed the amount of inbound message buffer space assigned to the inbound path. If the system operator did make a change resulting in less message buffer space for an inbound path, you should verify that there was a sound rationale for the action.

- Rule WLM620 is related to Rule WLM605. When Rule WLM605 is produced, CPExpert often will produce Rule WLM620 to identify the outbound/inbound path combination that is experiencing problems.

  - It is possible that Rule WLM605 would be produced but CPExpert cannot identify an outbound/inbound path combination causing problems. This situation could occur when there is a **general** problem with the inbound buffer space over all paths, but no path combination causes the problem.

  - It is possible that Rule WLM605 would **not** be produced, but CPExpert could produce Rule WLM620. This situation could occur when there is not a **general** problem with the inbound buffer space for all paths, but a particular outbound/inbound path is experiencing problems.

    Rule WLM605 is based on the PCTREJ guidance variable, which guides the assessment of rejects of outbound messages (analyzing SMF Type 74, Subtype 2, System Data). Rule WLM620 is based on comparing the outbound path BUSY with the inbound path BUFFER UNAVAILABLE condition (analyzing SMF Type 74, Subtype 2, Path Data).

    Since different data are analyzed by different logic paths, it is not always possible for CPExpert to produce both Rule WLM605 and Rule WLM620.

[1]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM621: Message buffer space may be too small for inbound path

Finding: CPExpert believes that the buffer space may be too small for an inbound path (XCF list structure).

Impact: This finding can have a MEDIUM IMPACT or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon how often XCF was unable to obtain buffer space for inbound messages. **This finding applies to XCF list structures**.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: This finding is similar to Rule WLM620, except this rule applies to list structures. Please refer to Rule WLM620 for a discussion and suggested alternatives.

The following example illustrates the output from Rule WLM621:

```
RULE WLM621: MESSAGE BUFFER SPACE MAY BE TOO SMALL FOR INBOUND PATH

   The OUTBOUND path busy was high for the IXCPLEX_PATH1 list structure
   from System JA0 to System J80, while the INBOUND path on System J80
   had a high BUFFER UNAVAILABLE condition.  This situation usually means
   that the list structure message buffer space should be increased.  The
   message buffer space was specified as 1000K for the IXCPLEX_PATH1
   list structure.  This finding applies to the following RMF measurement
   intervals:

                          OUTBOUND  PCT OUTBOUND  INBOUND   PCT  BUFFERS
   MEASUREMENT INTERVAL   REQUESTS  BUFFERS BUSY  REQUESTS   UNAVAILABLE
   12:30-13:00,26MAR1996    11,406      13.2        1,406         9.1
   13:00-13:30,26MAR1996    12,620      15.4        2,620         9.8
```

Rule WLM622: The number of outbound paths may need to be increased

Finding: The PATH BUSY (when selected for transfer) was high relative to the PATH AVAILABLE for the indicated path. CPExpert believes that outbound paths may need to be added to the indicated transport class.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon how often XCF was unable to obtain outbound paths when needed.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

Outbound paths are assigned to transport classes by using the CLASS parameter on the PATHOUT statement (or by using the SETXCF PATH command after IPL). At least one outbound signalling path should be assigned to each transport class[1]. If there is high message traffic in the transport class, you may wish to assign more than one signalling path to the transport class. Additionally, you may wish to assign more signalling paths for redundancy.

[1]You are not required to assign a signalling path to a transport class. If no signalling path is assigned to a transport class, the XCF groups in the transport class compete for signalling resources of other transport classes. This situation can degrade signalling performance.

©Copyright 1996, Computer Management Sciences, Inc.

XCF attempts to select signalling paths that can immediately transfer a message because these paths should provide the least amount of delay to the message.

SMF Type 74 (Subtype 2) provides statistics about the number of messages sent by XCF groups in a transport class, where the messages are sent, the paths used to send the messages, how BUSY each path was when selected for transfer, and how often each path was AVAILABLE when selected for transfer.

CPExpert analyzes this information to determine whether sufficient outbound paths have been defined. CPExpert evaluates the PATH BUSY versus PATH AVAILABLE for each outbound path. CPExpert concludes that the path is becoming overloaded when the PATH BUSY when selected for transfer was greater than 25% of the PATH AVAILABLE time.

The following example illustrates the output from Rule WLM622:

```
RULE WLM622: THE NUMBER OF OUTBOUND PATHS MAY NEED TO BE INCREASED

   The PATH BUSY (when selected for transfer) was high relative to the
   PATH AVAILABLE for the C605 path on System JB0, sending messages to the
   C611 path on System JA0 in transport class DEFSMALL.  This usually
   means that you need to add more OUTBOUND paths to the transport class.
   This finding applies to the following RMF measurement intervals:

                                        TOTAL     PCT OUTBOUND    PCT  PATH
   MEASUREMENT INTERVAL               MESSAGES     PATH  BUSY     AVAILABLE
   12:00-12:30,26MAR1996                 2562         21.1           78.9
```

Suggestion: If Rule WLM622 is regularly produced, CPExpert suggests that you consider the following alternatives[2]:

- You should evaluate the number of outbound paths specified for the transport class. You should examine the DEVICE parameter or the STRNAME parameter list of the PATHOUT statement associated with the transport class to determine whether additional paths should be assigned.

  In evaluating the number of paths assigned to the transport class, you should consider (1) the importance of the messages in the transport class, (2) how often the "ALL PATHS UNAVAILABLE" condition was experienced, and (3) the potential impact on other transport classes

[2]**WARNING**: There exists little practical experience with analyzing coupling facility data and with selecting proper values for the controlling parameters. The CPExpert analysis and suggestions are based on (1) the information contained in the referenced documents and (2) our analysis of data provided by IBM or CPExpert users. Please keep this paucity of knowledge in mind when considering the alternatives. Additionally, **please** provide Computer Management Sciences with feedback!

when XCF must route the outbound messages to paths assigned to other transport classes.

- You should evaluate whether the XCF groups are properly assigned to transport classes. XCF groups are assigned to transport classes via the GROUP parameter on the CLASSDEF statement.

  - XCF groups can be assigned to more than one transport class. When evaluating which transport class to use (when XCF groups are assigned to more than one transport class) XCF will select the transport class with the smallest buffer that will hold the message being sent. You potentially can "optimize" the buffer space used by assigning XCF groups to more than one transport class.

    All groups assigned to a transport class have equal access to the signalling resources of that class. Consequently, you should make sure that you do not assign "low priority" groups to transport classes that have high performance requirements if the "low priority" groups could cause performance degradation to the "high priority" groups.

    Fortunately, SMF Type 74 (Subtype 2) records contain information about the XCF groups and XCF members, including the number of signals sent and received by each member. This information is in the **Member Data Section** of the Type 74 records, and can be analyzed to assess the impact of message traffic of the XCF members and XCF groups.

  - Alternatively, it may be preferable to reassign XCF groups to transport classes. In practice, this situation is unlikely to occur as most installations will have a relatively small number of transport classes.

- If Rule WLM622 occurs frequently and there is no action you wish take, you can to exclude the transport class from CPExpert's analysis, using the **EXCLASSn** guidance variables. The EXCLASSn guidance variables allow you to exclude one or more transport classes from analysis.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM623: The number of outbound paths may need to be increased

Finding: The PATH BUSY (when selected for transfer) was high relative to the PATH AVAILABLE for the indicated path. CPExpert believes that outbound paths may need to be added to the indicated transport class. This finding applies to XCF list structures.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends upon how often XCF was unable to obtain outbound paths when needed.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: This finding is similar to Rule WLM622, except this rule applies to list structures. Please refer to Rule WLM622 for a discussion and suggested alternatives.

The following example illustrates the output from Rule WLM623:

```
RULE WLM623: THE NUMBER OF OUTBOUND PATHS MAY NEED TO BE INCREASED

   The PATH BUSY (when selected for transfer) was high relative to the
   PATH AVAILABLE for the IXCPLEX_PATH2 structure on System J90, sending
   messages to System J80 in transport class DEFSMALL.  This usually means
   that you need to add more OUTBOUND paths to the transport class.  This
   finding applies to the following RMF measurement intervals:

                                        TOTAL     PCT OUTBOUND    PCT  PATH
   MEASUREMENT INTERVAL               MESSAGES     PATH  BUSY     AVAILABLE
   10:30-11:00,26MAR1996                2,691         31.3           68.7
```

Suggestion: Please refer to Rule WLM622 for a discussion and suggested alternatives.

Rule WLM630: A hardware problem may exist

Finding: There was a significant number of PATH RETRY requests to one or more paths in the indicated transport class.

Impact: This finding can have a HIGH IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

When a signalling path experiences an error (such as an I/O error), XCF attempts to restart the signalling path and resend the message. Restarting a path represents a loss of signalling capacity while the path is being restarted. Additionally, the failed message must be resent on a different path during path restart and the delay to the message may cause sysplex performance degradation. Depending upon the type of message being sent, the performance degradation could be serious.

If the number of retry operations reaches the value specified for the RETRY parameter on the PATHIN or PATHOUT statement, XCF will stop the path. The default value for the RETRY parameter is 10, indicating that XCF will stop the path after 10 retry operations.

SMF Type 74 (Subtype 2) provides statistics about the number of messages, where the messages are sent, the paths used to send the

messages, the path retry limit (as specified in the PATHIN or PATHOUT statement), and how many retry operations XCF initiated for the path.

CPExpert analyzes this information to determine whether a hardware problem exists for the path. CPExpert concludes that a hardware problem exists in the path when the number of XCF retry operations was 25% of the path retry limit. The purpose of selecting this value is to give an "early warning" of pending path problems.

The following example illustrates the output from Rule WLM630:

```
RULE WLM630: A HARDWARE PROBLEM MAY EXIST

   There were a significant number of RETRY requests in the DEFSMALL
   transport class.  A RETRY often indicates that there is a hardware
   problem.  This finding applies to the following RMF measurement
   intervals:

                                 SENT TO       OUTBOUND                RESTART
   MEASUREMENT INTERVAL     (SYSTEM/PATH)    REQUESTS    RESTARTS     LIMIT
   10:30-11:00,26MAR1996       Z0/C594         6,245         26        100
   11:00-11:30,26MAR1996       Z0/C595         7,177         44        100
   11:30-12:00,26MAR1996       Z0/C596        10,508         63        100
   12:00-12:30,26MAR1996       Z0/C597        12,919         72        100
```

Suggestion: If Rule WLM630 is produced, CPExpert suggests that you identify and resolve the cause of the path retry problems.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"

Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM632: An inbound path was non-operational

Finding: CPExpert noticed that the indicated inbound path was non-operational.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or IMPACT on the signalling performance of the sysplex. The level of impact depends on the message traffic and the capacity of the inbound paths.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

An inbound path can be non-operational because of a hardware failure in which the number of time XCF had to retry the path was larger than the value of the RETRY parameter on the PATHIN statement. This condition results in an message to the operator (and CPExpert would generate Rule WLM630 if the hardware failure occurred during RMF intervals being analyzed).

A more insidious cause of a path being non-operational is that an error has been made in the path definition: an inbound path has been defined but no corresponding outbound path has been defined.

Alternatively, a system operator might have made an error:

- The operator could have issued a SETXCF START,PATHIN command to start an inbound path but did not issue a SETXCF START,PATHOUT command to start the corresponding outbound path on the other system.

- The operator could have issued a SETXCF DELETE,PATHOUT command to delete an outbound path but did not issue a SETXCF DELETE,PATHIN command to delete the corresponding inbound path on the other system.

In any of the above cases, the inbound path is defined to XCF, but XCF cannot use the path. This situation wastes resources and lowers the capacity of the signalling system.

SMF Type 74 (Subtype 2) provides statistics about the status of each path in the R742PSTA status flags:

Bit	**Meaning when set**[1]
0	Path starting
1	Path restarting
2	Path working
3	Path stopping
4	Path waiting for completion of initial protocol
5	Path not operational
6	Path stop failed
7	Path rebuilding
7	Path starting

CPExpert analyzes this information to determine whether a path has been defined to XCF but the path is not operational.

The following example illustrates the output from Rule WLM632:

Suggestion: If Rule WLM632 is produced, CPExpert suggests that you identify the reason the path is not operational.

- Rule WLM630 would have been produced if the path is not operational because of hardware problems, and the retry limit had been reached during the RMF intervals being analyzed. In this case, you should determine and correct the hardware problems.

[1]Please note that the SMF manual describes bits 5-7 as Reserved. Private communication with RMF developers at IBM, Germany revealed that bits 5-7 have the meaning shown above. The SMF manual will be updated with this information.

```
RULE WLM632: AN INBOUND PATH WAS NON-OPERATIONAL

   The C594 inbound path was non-operational during the following RMF
   measurement intervals.  The path was defined to XCF, but the path was
   not usable.  A path is not usable by XCF because of hardware problems,
   or because the path on the other end (the outbound path of another
   system) was not defined or was not defined correctly.

   MEASUREMENT INTERVAL
   10:00-10:30,26MAR1996
   10:30-11:00,26MAR1996
   11:00-11:30,26MAR1996
   11:30-12:00,26MAR1996
   12:00-12:30,26MAR1996
   12:30-13:00,26MAR1996
   13:00-13:30,26MAR1996
```

If there are no hardware problems with the path, you should review the signalling path definitions.

- Review the path definition in the PATHIN statement for the system identified. You should ensure that there is a corresponding PATHOUT statement for the other system.

- If the path definition in the PATHIN and PATHOUT statements are correct, you should review operator actions to ensure that the operator has taken proper action when starting or deleting a path. Either of the two situations described above (in the Discussion section) could cause an inbound path to be non-operational.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM633: An outbound path was non-operational

Finding: CPExpert noticed that the indicated outbound path was non-operational in the transport class described by this rule.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or IMPACT on the signalling performance of the sysplex. The level of impact depends on the message traffic and the capacity of the outbound paths assigned to the transport class.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: The XCF component of MVS/ESA allows authorized programs on one MVS system in a sysplex to communicate with programs on the same system or on other systems. A typical example of this communication is between CICS regions; CICS regions often communicate with other CICS regions in the same system or with CICS regions on other systems in the sysplex.

Please refer to the discussion associated with Rule WLM601 for additional information about XCF buffers.

XCF group members communicate with each other using the XCF *signalling* mechanism. The communication is done via signalling paths consisting of ESCON channels operating in channel-to-channel (CTC) mode, a coupling facility list structure (beginning with MVS/ESA Version 5), or 3088 Multisystem Channel Communication Unit. Messages are sent over the signalling paths, and the paths have one or more buffers associated with them to hold the messages as they are sent or received.

An outbound path can be non-operational because of a hardware failure in which the number of time XCF had to retry the path was larger than the value of the RETRY parameter on the PATHOUT statement. This condition results in an message to the operator (and CPExpert would generate Rule WLM630 if the hardware failure occurred during RMF intervals being analyzed).

A more insidious cause of a path being non-operational is that an error has been made in the path definition: an outbound path has been defined but no corresponding inbound path has been defined.

Alternatively, a system operator might have made an error:

- The operator could have issued a SETXCF START,PATHOUT command to start an outbound path but did not issue a SETXCF START,PATHIN command to start the corresponding inbound path on the other system.

- The operator could have issued a SETXCF DELETE,PATHIN command to delete an inbound path but did not issue a SETXCF DELETE,PATHOUT command to delete the corresponding outbound path on the other system.

In any of the above cases, the outbound path is defined to XCF, but XCF cannot use the path. This situation wastes resources and lowers the capacity of the signalling system.

SMF Type 74 (Subtype 2) provides statistics about the status of each path in the R742PSTA status flags:

Bit	**Meaning when set**[1]
0	Path starting
1	Path restarting
2	Path working
3	Path stopping
4	Path waiting for completion of initial protocol
5	Path not operational
6	Path stop failed
7	Path rebuilding

CPExpert analyzes this information to determine whether a path has been defined to XCF but the path is not operational.

The following example illustrates the output from Rule WLM633:

Suggestion: If Rule WLM633 is produced, CPExpert suggests that you identify the reason the path is not operational.

- Rule WLM630 would have been produced if the path is not operational because of hardware problems, and the retry limit had been reached during the RMF intervals being analyzed. In this case, you should determine and correct the hardware problems.

[1]Please note that the SMF manual describes bits 5-7 as Reserved. Private communication with RMF developers at IBM, Germany revealed that bits 5-7 have the meaning shown above. The SMF manual will be updated with this information.

```
RULE WLM632: AN INBOUND PATH WAS NON-OPERATIONAL

   The C594 inbound path was non-operational during the following RMF
   measurement intervals.  The path was defined to XCF, but the path was
   not usable.  A path is not usable by XCF because of hardware problems,
   or because the path on the other end (the outbound path of another
   system) was not defined or was not defined correctly.

   MEASUREMENT INTERVAL
   10:00-10:30,26MAR1996
   10:30-11:00,26MAR1996
   11:00-11:30,26MAR1996
   11:30-12:00,26MAR1996
   12:00-12:30,26MAR1996
   12:30-13:00,26MAR1996
   13:00-13:30,26MAR1996
```

If there are no hardware problems with the path, you should review the signalling path definitions.

- Review the path definition in the PATHOUT statement for the system identified. You should ensure that there is a corresponding PATHIN statement for the other system.

- If the path definition in the PATHIN and PATHOUT statements are correct, you should review operator actions to ensure that the operator has taken proper action when starting or deleting a path. Either of the two situations described above (in the Discussion section) could cause an outbound path to be non-operational.

Reference: MVS/ESA: Setting Up a Sysplex (GC28-1449)
Section 5: Planning Signalling Services in a Sysplex

MVS/ESA: Initialization and Tuning Reference (GC28-1452)
COUPLExx (Cross-System Coupling Facility Parameters)

OS/390: Setting Up a Sysplex (GC28-1779)
Section 5: Planning Signalling Services in a Sysplex

OS/390: Initialization and Tuning Reference (GC28-1752)
COUPLExx (Cross-System Coupling Facility Parameters)

z/OS: Setting Up a Sysplex (SA22-7625)
Section 5: Planning Signalling Services in a Sysplex

z/OS: Initialization and Tuning Reference (SA22-7592)
COUPLExx (Cross-System Coupling Facility Parameters)

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM651: Lock Contention was high for the indicated structure

Finding: The lock contention for the indicated structure was higher than guidance provided by IBM for normal lock contention.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends on the amount of lock contention.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: *Locking* is the mechanism used to reserve all or part of a database so that other programs will not be able to update the data until you have finished processing the data. By locking the data, users can be sure that the information they are processing is current. Without locking, users might lose updates or access invalid or incomplete data. Locking is necessary, of course, only if one or more of the users of the data will be performing updates. If no updating of the data is performed, locking is unnecessary; the data may be concurrently accessed by any number of user without worry that the data is incomplete or invalid.

Lock contention occurs when one user wishes to access data and some other user has placed a lock on the data. The user wishing to access the data usually is suspended until the data is available (that is, until the lock is released). Techniques such as separating data, choosing locking parameters, and monitoring for contention can be used to provide a balance between concurrency of access, isolation and integrity of data, and efficient use of system resources. Lock contention is analyzed by CPExpert in Rule WLM651.

SMF Type 74 (Subtype 4 - Coupling Facility Activity) records contain information describing the requests for data, the number of requests that were delayed because of lock contention, and the number of requests that encountered false lock contention. CPExpert analyzes this information to determine whether an excessive percentage of requests encountered lock contention.

CPExpert divides R744SSCN (the number of times any request encountered lock contention) by R744STRC (the total number of lock-related requests), to yield the percent of requests that experienced lock contention. CPExpert compares this percentage with the **LOCKCONT** guidance variable in USOURCE(WLMGUIDE).

©Copyright 1997, Computer Management Sciences, Inc.

CPExpert produces Rule WLM651 when the percent of lock contention exceeds the value specified by the LOCKCONT variable.

The default value for the LOCKCONT variable is 2%, indicating that CPExpert should produce Rule WLM651 when more than 2% of the requests were delayed because of lock contention. IBM documents suggest that most CICS/DBCTL workloads should have less than 1% lock contention, and most IMS/DB2 workloads should have less than 2% lock contention.
The following example illustrates the output from Rule WLM651:

```
RULE WLM651: LOCK CONTENTION WAS HIGH

   DB2DBP2_LOCK1:  The lock contention for this structure was higher than
   normal.  High lock contention can result in an increase in central
   processor utilization and a reduction in throughput.  If this finding
   continues to occur, you should review the alternatives listed in the
   WLM Component User Manual.  If you are unable to take action, you
   should consider increasing the LOCKCONT guidance variable, located
   in USOURCE(WLMGUIDE).  The LOCKCONT variable currently is 2%.

                                  TOTAL LOCK   REQUESTS WITH   PERCENT LOCK
   MEASUREMENT INTERVAL            REQUESTS   LOCK CONTENTION   CONTENTION
   12:45-13:00,02OCT1996            17,696           910             5
   16:45-17:00,02OCT1996            12,320           757             6
   17:15-17:30,02OCT1996             3,741           371            10
```

Suggestion: A frequent cause of high lock contention is batch jobs running against shared databases. If this is the cause, you may be able to reschedule the batch jobs to resolve the lock contention problem.

Additionally, CPExpert suggests that you consider the following alternatives, depending on the type of lock structure experiencing the contention:

- If the structure involved is CICS/DBCTL, you should refer to the CICS-IMS DBCTL Guide for a discussion about lock contention and suggestions on how to prevent the lock contention. IBM provides the following recommendations on ways to reduce lock contention:

  - All BMPs and applications should issue frequent checkpoints to avoid locking out other resource users.

  - All BMPs and applications must be restartable from last checkpoint. This is because records in the same database may have since been updated, and these updates would be lost if the database were restored from a previous backup.

- BMPs and applications should not hold on to locks for long periods without issuing checkpoints or syncpoints (either explicitly or implicitly).

- Review the use of control records; that is, records that are accessed by most applications. If they have to be updated, it is important to remember that the CI or physical block is locked from other subsystems until the updates are committed.

- If the structure involved is DB2, you should refer to the following sections of the indicated documents (shown in the References section of this rule description) for the appropriate version of DB2 running on your system:

  - "Tuning your use of Locks" section and the "Improving Concurrency" section in the Data Sharing: Planning and Administration document.

  - "Improving Concurrency" section in the DB2 Administration Guide

  - "Archive Log Data Set Parameters" section in the DB2 Installation Guide

  If you decide that the DB2 application design is causing lock contention, you should refer to the DB2 Application Programming and SQL Guide for detailed suggestions about how to avoid or minimize lock contention.

- If the structure involved is JES2 checkpoint, the structure will be a *serialized list structure* with locking controlled by JES2 in a multi-access spool (MAS) environment.

  Each JES2 member of the MAS will acquire and hold the checkpoint for the duration specified in the HOLD parameter on the MASDEF initialization statement. Upon releasing the lock on the checkpoint, each JES2 member of the MAS will wait for the interval specified in the DORMANCY parameter on the MASDEF initialization statement before attempting to again acquire the checkpoint.

  Please refer to the "Accessing the CKPTn Data Set in a MAS" section of the JES2 Initialization and Tuning Guide for IBM's suggestions on setting the HOLD and DORMANCY parameters in a MAS environment in which the CKPTn data sets reside in structures on a coupling facility.

©Copyright 1997, Computer Management Sciences, Inc.

Reference: OS/390: DB2 for MVS/ESA Version 4
Data Sharing: Planning and Administration (SC26-3269)
Administration Guide (SC26-3265)
Installation Guide (SC26-3456)
Application Programming and SQL Guide (SC26-3266)

OS/390: DB2 for MVS/ESA Version 5
Data Sharing: Planning and Administration (SC26-8961)
Administration Guide (SC26-8957)
Installation Guide (SC26-8970)
Application Programming and SQL Guide (SC26-8958)

OS/390: DB2 for MVS/ESA Version 6
Data Sharing: Planning and Administration (SC26-9007)
Administration Guide (SC26-9003)
Installation Guide (SC26-9008)
Application Programming and SQL Guide (SC26-9004)

OS/390: JES2 Initialization and Tuning Guide (SC28-1791)

OS/390: RMF Performance Management Guide (SC28-1951)

z/OS: JES2 Initialization and Tuning Guide (SA28-7532)

z/OS: RMF Performance Management Guide (SC33-7992)

Washington System Center Flash 9609 ("CF Reporting Enhancements to RMF 5.1")

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

©Copyright 1997, Computer Management Sciences, Inc.

Rule WLM652: False Lock Contention was high for the indicated structure

Finding: The false lock contention for the indicated structure was higher than guidance provided by IBM.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or IMPACT on the signalling performance of the sysplex. The level of impact depends on the amount of false lock contention. However, when analyzing the impact of this finding, you should keep in mind that (1) false lock contention requires overhead and (2) false lock contention is unnecessary as it normally can be eliminated.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: *Locking* is the mechanism used to reserve all or part of a database so that other programs will not be able to update the data until the user placing the lock has finished processing the data. By locking the data, users can be sure that the information they are processing is current. Without locking, users might lose updates or access invalid or incomplete data. Locking is necessary, of course, only if one or more of the users of the data will be performing updates. If no updating of the data is performed, locking is unnecessary; the data may be concurrently accessed by any number of user without worry that the data is incomplete or invalid.

Lock contention occurs when one user wishes to access data and some other user has placed a lock on the data. The user wishing to access the data usually is suspended until the data is available (that is, until the lock is released). Techniques such as separating data, choosing locking parameters, and monitoring for contention can be used to provide a balance between concurrency of access, isolation and integrity of data, and efficient use of system resources..

The coupling facility lock structure contains information used to determine cross-system contention on a particular resource. IRLM assigns (or "hashes") locked resources to an entry value in the lock structure in the coupling facility. IRLM uses the lock table to determine whether a resource is locked. If the lock structure defined on the coupling facility is too small, the hashing algorithm can select the same lock table entry for two different locks. This situation is termed *false lock contention*. The user wishing to access the locked data is suspended until it is determined that there is no real lock contention on the resource.

©Copyright 1997, Computer Management Sciences, Inc.

SMF Type 74 (Subtype 4 - Coupling Facility Activity) records contain information describing the requests for data, the number of requests that were delayed because of lock contention, and the number of requests that encountered false lock contention. CPExpert analyzes this information to determine whether an excessive percentage of requests encountered false lock contention.

IBM documents have been inconsistent with respect to guidance for false lock contention.

- IBM's *Setting up a Sysplex* document contained Section 6.3.1: Lock Contention prior to OS/390 Version 2 Release 6. This section specified an objective that no more than 0.1% of **total** requests should experience false lock contention. This section was removed completely with OS/390 Version 2 Release 6 with the comment in the CHANGES section that the document "has been updated with more recent information about tuning coupling facilities. No guidance about excessive false lock contention is contained in the *Setting up a Sysplex* document after V2R5.

- With DB2 Version 4, Section 6.3.2.3: Avoid False Contention, IBM stated "If possible, try to keep false contention to no more than 50 percent of total global lock contention. (However, if total global lock contention is a very low value, it might not be as necessary to reduce false contention.)"

- With DB2 Version 5, IBM removed that statement from Section 6.3.2, and provided no specific guidance regarding false lock contention. Rather, IBM calculated false contention as "the number of false contentions divided by the total number of requests that went to XES (excluding asynchronous requests)." That particular calculation in the example given by IBM resulted in 0.4, which IBM concluded "false contention is 0.4 percent, a very good number."

- *DFSMSdfp Storage Administration Reference* (SC26-7331) for OS/390 V2R10 contains the statement: "A good goal is to have total (real and false) global lock contention of less than one percent. The false contention component of the total global lock contention should be less than one-half of one percent, and ideally, should be substantially less than this." Additionally, the discussion on defining a lock structure states "These lock size estimates include the memory requirements for both the lock table and the record-lock memory. Use these estimates as rough initial values to help you attain a locking structure with a desired false contention target of approximately one-half of 1% or less. "

- OS/390: RMF Performance Management Guide (SC28-1951) still contains the “no more than 0.1% of **total** requests” statement, but this document likely has not been not regularly updated.

CPExpert divides R744SFCN (the number of times any request encountered false lock contention) by R744STRC (the total number of lock-related requests) for lock structures, to yield the percent of requests that experienced false lock contention. CPExpert produces Rule WLM652 when this percent is more than the value specified for the **FALSECNT** guidance variable.

The default value for the **FALSECNT** guidance variable is 0.5%, indicating that CPExpert should produce Rule WLM652 when more than one-half of one percent of the lock-related requests encountered false lock contention.

CPExpert additionally checks that the overall lock contention was at least as high as 25% of the value specified in the **LOCKCONT** guidance variable. This test is made to avoid spurious production of Rule WLM652.

The following example illustrates the output from Rule WLM652:

```
RULE WLM652: FALSE LOCK CONTENTION WAS HIGH

   DB2DBP2_LOCK1:  The number of locks with false contention should be less
   than 0.5% of the total requests. The false lock contention exceeded the
   guideline for this structure.  False lock contention occurs when the
   hashing algorithm hashes to the same lock table entry (hash value) for
   two different locks.  False lock contention can be reduced by increasing
   the size of the lock structure or selecting a better value for the
   MAXUSRS parameter in IRLMPROC.  Refer to Rule WLM652 in the WLM Component
   User Manual for additional suggestions.

                                TOTAL LOCK    FALSE LOCK     PERCENT FALSE
   MEASUREMENT INTERVAL          REQUESTS     CONTENTION    LOCK CONTENTION
   15:15-15:30,02OCT1996          12,676         2,654             21
```

Suggestion: False lock contention often can be reduced by increasing the size of the lock structure or selecting a better value for the MAXUSRS parameter in IRLMPROC. Please note that if you do increase the size of the lock structure, you should increase by a power of 2 to avoid wasting storage.

Additionally, CPExpert suggests that you consider the following alternatives, depending on the type of lock structure experiencing the contention:

- If the structure involved is DB2, you should refer to the "Avoiding False Lock Contention" section in the DB2 Data Sharing: Planning and Administration document for your version of DB2.

- If you are using VSAM Record Level Sharing (RLS), you should refer to "Avoiding False Lock Contention" in the DFSMSdfp Storage Administration Reference.

- You can adjust CPExpert's analysis by altering the value specified for the **PCTFALSE** guidance variable in USOURCE(WLMGUIDE).

Reference: OS/390: Setting Up a Sysplex (GC28-1779) for OS/390 prior to V2R6
Section 6.3.1: Lock Contention

OS/390: DB2 for MVS/ESA Version 4 Data Sharing: Planning and Administration (SC26-3269)
Section 6.3.2.3: Avoid False Lock Contention

OS/390: DB2 for MVS/ESA Version 5 Data Sharing: Planning and Administration (SC26-8961)
Section 7.4.2.3: Avoid False Lock Contention

OS/390: DB2 for MVS/ESA Version 6 Data Sharing: Planning and Administration (SC26-9007)
Section 7.5.2.3: Avoid False Lock Contention

OS/390 and z/OS: DB2 for MVS/ESA Version 7 Data Sharing: Planning and Administration (SC26-9935)
Section 6.5.2.2: Avoid False Lock Contention

DFSMSdfp Storage Administration Reference for OS/390 (SC26-7331)
Section 14.1.8.2: Avoiding False Contention

OS/390: RMF Performance Management Guide (SC28-1951)
Section 6.4.4.2: Don't Make Additional Work

z/OS: RMF Performance Management Guide (SC33-7992)
Section 6.2.4.2: Don't Make Additional Work

Rule WLM660: Synchronous service time was high for the indicated structure

Finding: The synchronous service time for the indicated structure exceeded the guidance provided to CPExpert.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends on the amount of delay to synchronous requests and how important the requests are.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Signalling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available. The type of action taken by XES depends on whether the request was specified as synchronous or asynchronous.

- Synchronous requests require that a response be received from the coupling facility before the requesting application continues execution. Synchronous requests would be used, for example, to request a lock. In this example, the application cannot proceed until the lock is granted.

  For synchronous requests, XES will either (1) satisfy the request if a subchannel is available, (2) enter CPU "spin-looping" until a subchannel is available and the request is satisfied, or (3) convert the synchronous request to an asynchronous request if the type of request permits the conversion.

- Asynchronous requests allow the requesting application to continue processing and be notified when the request is completed. For asynchronous requests, XES either starts or queues the request and returns control to the application issuing the request.

The type (synchronous or asynchronous) of request that is issued generally depends on the type of structure.

- Some requests can be satisfied only by synchronous requests (for example, signals generated by XES itself will always be synchronous and will not be converted to asynchronous requests).

©Copyright 1997, Computer Management Sciences, Inc.

- Some requests can be issued as either synchronous or asynchronous requests, depending on the application's use of the structure (for example, cache structure requests can be issued as either synchronous or asynchronous).

- Some requests are issued as asynchronous requests (for example, JES2 requests to the JES2 checkpoint will be issued as asynchronous requests).

- Some requests can be issued as synchronous but will be converted to asynchronous if the subchannels are busy[1] unless the application has indicated that the synchronous cannot be converted.

The time spent waiting for subchannels to become free for synchronous requests not only delays the request (and consequently delays the application waiting on the request), but wastes processor resources since the processor is in a CPU "spin-loop" waiting for the synchronous request to be satisfied.

The service time represents the time from when MVS issues a command for the coupling facility until the return from the command is recognized by MVS. The time includes time spent on the coupling facility links, the coupling facility processing time, and the time for MVS to recognize that the command was completed. The service time varies based on whether subchannels are available, the activity level of the coupling facility itself, and on the amount of data being processed.

IBM suggests that the service time for synchronous requests should be less than 250-350 microseconds, depending upon the length of the request. The service time for lock structures should be less than 250 microseconds, since lock structure requests are small.

CPExpert compares the synchronous service time (R744SSTM) against the **SYNCSRV** variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM660 when the synchronous service time is greater than the SYNCSRV guidance variable.

The default value for the SYNCSRV variable is 350, indicating that CPExpert should produce Rule WLM660 when synchronous service time is more than 350 microseconds. CPExpert subtracts 100 microseconds from the SYNCSRV guidance variable if evaluating a lock structure. Thus, CPExpert will produce Rule WLM660 when the service time for lock structures is greater than 250 microseconds.

---

[1]The application can specify which requests must be satisfied as synchronous and which can be converted to asynchronous. XES will automatically convert requests from synchronous to asynchronous if all signalling paths are busy, unless the application specifies that the conversion is not to be done.

The following example illustrates the output from Rule WLM660:

```
RULE WLM660: SERVICE TIME WAS HIGH FOR SYNCHRONOUS REQUESTS

   ISGLOCK:  The service time for this structure has exceeded the
   guidelines for synchronous requests.  Service time is accumulated from
   the time MVS issues a command for the coupling facility until the return
   from the command is recognized by MVS.  Service time is recorded for
   each structure used by each system.  You can alter the times used by
   CPExpert in making this finding by altering the SYNCSRV guidance
   variables in USOURCE(MVSGUIDE) if you are unable to make changes to
   reduce service time for the structure.

                                 TOTAL SYNC      AVERAGE SERVICE
   MEASUREMENT INTERVAL           REQUESTS     TIME (MICROSECONDS)
   10:45-11:00,06MAR1997           2,052                9,104
```

Suggestion: CPExpert suggests that you consider the following alternatives if Rule WLM660 is produced:

- Synchronous command processing is performed by the CP. You should make certain that sufficient CPU resources have been allocated to the coupling facility LPAR.

- Examine whether the structure activity is balanced between coupling facilities. You may wish to consider redistributing the structures among the coupling facilities if a significant imbalance exists.

- You should consider whether additional coupling facility links should be added between the MVS processor the coupling facility. Each coupling facility link will contribute two subchannels.

- If possible, you should consider influencing the exploiters of the coupling facilities to lower the activity rate to the coupling facilities. Taking other tuning actions (especially if indicated by other rules produced by CPExpert) may reduce the number of XCF signals. For example, signal activity can be lowered by (1) reducing lock contention, (2) reducing false lock contention, or (3) tuning the XCF to eliminate signals related to the expansion of a transport class size.

- If none of the above alternatives are appealing, you may wish to change the guidance to CPExpert by altering the **SYNCSRV** guidance variable in USOURCE(WLMGUIDE).

  IBM provides the following example service times based on measurements of CF lock requests for various combinations of sender CPCs and CFs. These measurements were reported in *S/390 MVS*

Parallel Sysplex Configuration, Volume 2: Cookbook, document Number SG24-2076-00.

Central Processor	**Coupling facility**	**Service Time** (microseconds)
9672R1 based	9674C01	250
9672R2/R3 based	9674C02/3	180
9672G3	9674C04	140
9672G4	9674C04	100
9672G4	9674C05	70
9021 711 based	9674C01	160
9021 711 based	9674C02/3	130
9021 711 based	9674C04	100
9021 711 based	9021 711 based	80

Reference: Washington System Center Flash 9609 ("CF Reporting Enhancements to RMF 5.1")

S/390 MVS Parallel Sysplex Configuration, Volume 2: Cookbook, document Number SG24-2076-00.

"Parallel Sysplex Performance: tuning tips and techniques," Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM661: Asynchronous service time was high for the indicated structure

Finding: The asynchronous service time for the indicated structure exceeded the guidance provided to CPExpert.

Impact: This finding can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex. The level of impact depends on the amount of delay to asynchronous requests and how important the requests are.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Signalling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available. The type of action taken by XES depends on whether the request was specified as synchronous or asynchronous.

- Synchronous requests require that a response be received from the coupling facility before the requesting application continues execution. Synchronous requests would be used, for example, to request a lock. In this example, the application cannot proceed until the lock is granted.

  For synchronous requests, XES will either (1) satisfy the request if a subchannel is available, (2) enter CPU "spin-looping" until a subchannel is available and the request is satisfied, or (3) convert the synchronous request to an asynchronous request if the type of request permits the conversion.

- Asynchronous requests allow the requesting application to continue processing and be notified when the request is completed. For asynchronous requests, XES either starts or queues the request and returns control to the application issuing the request.

The type (synchronous or asynchronous) of request that is issued generally depends on the type of structure.

- Some requests can be satisfied only by synchronous requests (for example, signals generated by XES itself will always be synchronous and will not be converted to asynchronous requests).

©Copyright 1997, Computer Management Sciences, Inc.

- Some requests can be issued as either synchronous or asynchronous requests, depending on the application's use of the structure (for example, cache structure requests can be issued as either synchronous or asynchronous).

- Some requests are issued as asynchronous requests (for example, JES2 requests to the JES2 checkpoint will be issued as asynchronous requests).

- Some requests can be issued as synchronous but will be converted to asynchronous if the subchannels are busy[1] unless the application has indicated that the synchronous cannot be converted.

The time spent waiting for subchannels to become free for asynchronous requests delays the request (and consequently delays the application waiting on the request).

The service time represents the time from when MVS issues a command for the coupling facility until the return from the command is recognized by MVS. The time includes time spent on the coupling facility links, the coupling facility processing time, any delay time while the request is queued, and the time for MVS to recognize that the command was completed. The service time varies based on whether subchannels are available, the activity level of the coupling facility itself, and on the amount of data being processed.

IBM suggests that the service time for asynchronous requests should be less than 5000 microseconds.

CPExpert compares the asynchronous service time (R744ASTM) against the **ASYNCSRV** variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM661 when the asynchronous service time is greater than the ASYNCSRV guidance variable.

The default value for the ASYNCSRV variable is 5000, indicating that CPExpert should produce Rule WLM661 when asynchronous service time is more than 5000 microseconds.

The following example illustrates the output from Rule WLM661:

[1]The application can specify which requests must be satisfied as synchronous and which can be converted to asynchronous. XES will automatically convert requests from synchronous to asynchronous if all signalling paths are busy, unless the application specifies that the conversion is not to be done.

```
RULE WLM661: SERVICE TIME WAS HIGH FOR ASYNCHRONOUS REQUESTS

   DB2DBP2_GBP2:  The service time for this structure has exceeded the
   guidelines for asynchronous requests.  Service time is accumulated from
   the time MVS issues a command for the coupling facility until the return
   from the command is recognized by MVS.  Service time is recorded for
   each structure used by each system.  You can alter the times used by
   CPExpert in making this finding by altering the ASYNCSRV guidance
   variables in USOURCE(WLMGUIDE) if you are unable to make changes to
   reduce service time for the structure.

                                TOTAL  ASYNC      AVERAGE SERVICE
   MEASUREMENT INTERVAL           REQUESTS      TIME (MILLISECONDS)
   12:45-13:00,02OCT1996             154                5.91
   13:00-13:15,02OCT1996              95                7.11
   14:00-14:15,02OCT1996             156                5.52
   15:45-16:00,02OCT1996              53                6.04
   16:45-17:00,02OCT1996             167                5.57
```

Suggestion: CPExpert suggests that you consider the following alternatives if Rule WLM661 is produced:

- Asynchronous command processing is performed primarily by the I/O processor. You should make certain that sufficient CPU resources have been allocated to the coupling facility LPAR.

- Examine whether the structure activity is balanced between coupling facilities. You may wish to consider redistributing the structures among the coupling facilities if a significant imbalance exists.

- You should consider whether additional coupling facility links should be added between the MVS processor the coupling facility. Each coupling facility link will contribute two subchannels.

- If possible, you should consider influencing the exploiters of the coupling facilities to lower the activity rate to the coupling facilities. Taking other tuning actions (especially if indicated by other rules produced by CPExpert) may reduce the number of XCF signals. For example, signal activity can be lowered by (1) reducing lock contention, (2) reducing false lock contention, or (3) tuning the XCF to eliminate signals related to the expansion of a transport class size.

If none of the above alternatives are appealing, you may wish to change the guidance to CPExpert by altering the **ASYNCSRV** guidance variable in USOURCE(WLMGUIDE).

Reference: Washington System Center Flash 9609 ("CF Reporting Enhancements to RMF 5.1")

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM665: Too many synchronous requests were changed to asynchronous requests

Finding: An unacceptably large number of synchronous requests were changed to asynchronous requests.

Impact: This finding can have a MEDIUM IMPACT, or HIGH IMPACT on the signalling performance of the sysplex.

Logic flow: This a basic finding. There are no predecessor rules.

Discussion: Signalling requests to a coupling facility can occur only if a subchannel to the coupling facility is available. If no subchannel is available, the cross-system extended services (XES) will either enter a CPU "spin loop" waiting for a subchannel to become available or queue the request until a subchannel is available. The type of action taken by XES depends on whether the request was specified as synchronous or asynchronous.

- Synchronous requests require that a response be received from the coupling facility before the requesting application continues execution. Synchronous requests would be used, for example, to request a lock. In this example, the application cannot proceed until the lock is granted.

  For synchronous requests, XES will either (1) satisfy the request if a subchannel is available, (2) enter CPU "spin-looping" until a subchannel is available and the request is satisfied, or (3) convert the synchronous request to an asynchronous request if the type of request permits the conversion.

- Asynchronous requests allow the requesting application to continue processing and be notified when the request is completed. For asynchronous requests, XES either starts or queues the request and returns control to the application issuing the request.

- Some requests can be issued as synchronous but will be converted to asynchronous if the subchannels are busy[1] unless the application has indicated that the synchronous cannot be converted.

[1]The application can specify which requests must be satisfied as synchronous and which can be converted to asynchronous. XES will automatically convert requests from synchronous to asynchronous if all signalling paths are busy, unless the application specifies that the conversion is not to be done.

©Copyright 1997, Computer Management Sciences, Inc.

There is a significant overhead associated with changing synchronous requests to asynchronous requests. XES must initially detect that the synchronous request is not going to be satisfied, the request must be changed to asynchronous, the request is queued, XES must detect when a subchannel is available, de-queue the asynchronous request, and process the asynchronous request. Not only is this overhead expensive in terms of resource consumption, but it is expensive in terms of delay to the application issuing the synchronous request.

The number of synchronous requests changed to asynchronous should be very low, to minimize the overhead and the delay to applications. IBM suggests that action should be taken when more than 10% of the synchronous requests are changed to asynchronous requests. This percentage is, obviously, dependent upon the application and the importance of the requests.

CPExpert computes the percent of synchronous requests changed to asynchronous requests (R744SSTA/R744SSRC). CPExpert produces Rule WLM665 when this percent is greater than the SYNCCHG guidance variable.

The default value for the SYNCCHG guidance variable is 10, indicating that CPExpert should produce Rule WLM665 when more than 10% of the synchronous requests are changed to asynchronous requests.

The following example illustrates the output from Rule WLM665:

```
RULE WLM665: TOO MANY SYNCHRONOUS REQUESTS WERE CHANGED TO ASYNCHRONOUS

   DB2DBP2_GBP3:  The structure experienced too many requests being changed
   from synchronous to asynchronous.  If MVS determines that a synchronous
   request will be significantly delayed (perhaps because subchannels are
   busy), MVS will change the request to an asynchronous request (note that
   synchronous lock requests are not changed).  This finding could indicate
   that you need additional coupling facility links.

                                TOTAL SYNC   SYNCH REQUESTS     PERCENT
   MEASUREMENT INTERVAL          REQUESTS   CHANGED TO ASYNCH   CHANGED
    7:15- 7:30,03OCT1996            242             29           12.0
```

Suggestion: Changed requests normally are caused by subchannel unavailable conditions. CPExpert suggests that you consider the following alternatives if Rule WLM665 is produced:

- You should make certain that sufficient CPU resources have been allocated to the coupling facility LPAR.

- You should consider whether additional coupling facility links should be added between the MVS processor the coupling facility. Each coupling facility link will contribute two subchannels.

- You should examine whether the structure activity is balanced between coupling facilities. You may wish to consider redistributing the structures among the coupling facilities if a significant imbalance exists.

- If possible, you should consider influencing the exploiters of the coupling facilities to lower the activity rate to the coupling facilities. Taking other tuning actions (especially if indicated by other rules produced by CPExpert) may reduce the number of XCF signals. For example, signal activity can be lowered by (1) reducing lock contention, (2) reducing false lock contention, or (3) tuning the XCF to eliminate signals related to the expansion of a transport class size.

If none of the above alternatives are appealing, you may wish to change the guidance to CPExpert by altering the **SYNCCHG** guidance variable in USOURCE(WLMGUIDE).

Reference: Washington System Center Flash 9609 ("CF Reporting Enhancements to RMF 5.1")

"Parallel Sysplex Performance: tuning tips and techniques,"
Kelley, Joan (IBM, Poughkeepsie, NY), SHARE 86, February 1996.

Rule WLM701: Coupling facility log stream structure was full

Finding: The SMF Type 88 records showed that a log stream coupling facility structure experienced a “structure full” condition.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 records.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. The system logger component resides in its own address space on each system in a sysplex. Applications can log data from one system or from multiple systems across the sysplex.

Applications write log data into a *log stream*. From the MVS view, the log stream is a set of records in time sequence order, merged into a single stream, independent of physical residence of the log stream. The log stream can reside in data space storage, in a staging data set, in a coupling facility, or in a log stream DASD data set. System parameters control the placement and length of log stream.

Applications that use the system logger services include:

- **Logrec**. Logrec log stream is an MVS system logger application that records hardware failures, selected software errors, and selected system conditions across the sysplex.

- **Operations log (OPERLOG)**. OPERLOG is an MVS system logger application that records and merges messages about programs and system functions (the hard copy message set) from each system in a sysplex that activates OPERLOG.

- **CICS Log Manager with CICS/Transaction Server for OS**/390. CICS log manager is a CICS system logger application that replaces the journal control management function.

- **IMS Common Queue Server Log Manager**. IMS common shared queues (CQS) log manager is a system logger application that records the information necessary for CQS to recover structures and restart.

©Copyright 1995, Computer Management Sciences, Inc.

- **APPC/MVS**. APPC/MVS is an MVS system logger application that records events related to protected conversations.

- **RRS (resource recovery services)**. RRS is an MVS system logger application that records events related to protected resources.

One significant advantage of the MVS system logger design is that any other system in a sysplex can recover data in the log stream. This feature prevents data loss in case of failure of one system.

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Version Release 2.4 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

- For a log stream that uses a coupling facility structure, a 'STRUCTURE FULL' condition can exist. In this case, the coupling facility has reached its capacity before off loading data to DASD[1]. This condition is analyzed by Rule WLM701.

- For a DASD-only log stream or for a log stream that is duplexed to a staging data set, a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set has reached its capacity before off loading data to secondary storage. This condition is analyzed by Rule WLM702.

If either of the above situations occur, they indicate that the logger cannot write data to secondary storage quickly enough to keep up with incoming data. Once the coupling facility space for a log stream is filled, system logger rejects all write requests until the coupling facility log data can be offloaded to DASD log data sets. Both situations can cause the application to wait before it can write more data. Depending on the length of time the application must wait, significant performance degradation would be experienced.

CPExpert examines the SMF88STN variable in the MXG TYPE88 data set (this variable indicates whether the log stream is a coupling facility type, or is a DASDONLY type). When this variable indicates the log stream is a coupling facility type, CPExpert compares the SMF88ESF (times a structure full condition was detected) variable in the MXG TYPE88 data set with the **STRFULL** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM701 when the SMF88ESF value exceeds the **STRFULL** guidance variable. The default value for the **STRFULL**

[1]This condition could be encountered during the rebuilding of a coupling facility structure, but rebuilding of a coupling facility structure is an event that would not require CPExpert's analysis - such an event would be well-known to systems personnel!

guidance variable is zero, indicating that CPExpert should produce Rule WLM701 when any structure full condition was detected.

Suggestion: IBM suggests that you consider the following alternatives to reduce the structure full conditions:

- Increase the size of the coupling facility structure in order to smooth out spikes in logger load.
- Reduce the HIGHOFFLOAD threshold percentage (the point at which the system logger begins off loading data from primary storage to off-load data sets).
- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. CPExpert normally will produce Rule WLM707 if too many DASD shifts occurred.
- Examine device I/O statistics for possible contention on the I/O subsystem used for off-load data sets.
- Use faster DASD devices.
- For CICS log streams, reduce the data written to the log stream by not merging so many journals or forward recovery logs onto the same stream.

Reference: OS/390 MVS: Setting up a Sysplex

OS/390 (V2R2):	Section 9.2.4
OS/390 (V2R3):	Section 9.2.5
OS/390 (V2R4):	Section 9.2.6
OS/390 (V2R5):	Section 9.2.6
OS/390 (V2R6):	Section 9.2.6
OS/390 (V2R7):	Section 9.2.6
OS/390 (V2R8):	Section 9.2.6
OS/390 (V2R9):	Section 9.2.6
OS/390 (V2R10):	Section 9.2.6
z/OS (V1R1):	Section 9.2.6
z/OS (V1R2):	Section 9.2.6, Section 9.4.5
z/OS (V1R3):	Section 9.2.6, Section 9.4.5
z/OS (V1R4):	Section 9.2.6, Section 9.4.5

©Copyright 1995, Computer Management Sciences, Inc.

Rule WLM702: Log stream staging data set was full

Finding: The SMF Type 88 records showed that a log stream staging data set experienced a “staging data set full” condition.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 records.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. The system logger component resides in its own address space on each system in a sysplex. Applications can log data from one system or from multiple systems across the sysplex.

Prior to OS/390 Release 2.4, the MVS system logger required a coupling facility (unless appropriate APARs were installed with OS/390 Release 1.3). With OS/390 Version Release 2.4 (or OS/390 Release 1.3 with appropriate APARs), individual log streams can use either DASD or a coupling facility.

- For a log stream that uses a coupling facility structure, a 'STRUCTURE FULL' condition can exist. In this case, the coupling facility has reached its capacity before off loading data to DASD[1]. This condition is analyzed by Rule WLM701.

- For a DASD-only log stream or for a log stream that is duplexed to a staging data set , a 'STAGING DATA SET FULL' condition can exist. In this case, the staging data set has reached its capacity before off loading data to secondary storage. This condition is analyzed by Rule WLM702.

If either of the above situations occur, they indicate that the logger cannot write data to secondary storage quickly enough to keep up with incoming data. Once the staging data set space for a log stream is filled, system logger rejects all write requests until the staging data set log data can be offloaded to DASD log data sets. Both situations can cause the application to wait before it can write more data. Depending on the length of time the

[1]This condition could be encountered during the rebuilding of a coupling facility structure, but rebuilding of a coupling facility structure is an event that would not require CPExpert’s analysis - such an event would be well-known to systems personnel!

©Copyright 1995, Computer Management Sciences, Inc.

application must wait, significant performance degradation would be experienced.

CPExpert compares the SMF88ETF (times a staging data set full was detected) variable in the MXG TYPE88 data set with the **LGDSFULL** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM702 when the SMF88ETF value exceeds the **LGDSFULL** guidance variable. The default value for the **LGDSFULL** guidance variable is zero, indicating that CPExpert should produce Rule WLM702 when any staging data set full condition was detected.

Suggestion: IBM suggests that you consider the following alternatives to reduce the staging data set full conditions:

- Increase the size of the staging data set.
- Reduce the HIGHOFFLOAD threshold percentage (the point at which the system logger begins off loading data from primary storage to off-load data sets).
- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. CPExpert normally will produce Rule WLM707 if too many DASD shifts occurred.
- Examine device I/O statistics for possible contention on the I/O subsystem used for off-load data sets.
- Use faster DASD devices.
- For CICS log streams, reduce the data written to the log stream by not merging so many journals or forward recovery logs onto the same stream.

Reference: OS/390 MVS: Setting up a Sysplex

OS/390 (V2R6):	Section 9.2.6, Section 9.4.5
OS/390 (V2R7):	Section 9.2.6, Section 9.4.5
OS/390 (V2R8):	Section 9.2.6, Section 9.4.5
OS/390 (V2R9):	Section 9.2.6, Section 9.4.5
OS/390 (V2R10):	Section 9.2.6, Section 9.4.5
z/OS (V1R1):	Section 9.2.6, Section 9.4.5
z/OS (V1R2):	Section 9.2.6, Section 9.4.5
z/OS (V1R3):	Section 9.2.6, Section 9.4.5
z/OS (V1R4):	Section 9.2.6, Section 9.4.5

Rule WLM703: Log stream structure offloads occurred: 90% full

Finding: The SMF Type 88 data showed that log stream structure offloads occurred because the structure was 90% full. This finding applies only to log streams that are defined to use a coupling facility structure.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is contained in coupling facility list structures.
- In a DASD-only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are

©Copyright 1995, Computer Management Sciences, Inc.

expressed as a percent of the interim storage[1] being filled. For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure is filled to the **HIGHOFFLOAD threshold** point or beyond, the system logger begins offloading data from the coupling facility to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **LOWOFFLOAD threshold** is the point in the coupling facility structure, as a percent of space consumed, where the system logger stops offloading log stream data to DASD log data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

From the above description, the amount of data that normally is offloaded is the difference between HIGHOFFLOAD and LOWOFFLOAD, as percentages of the coupling facility list structure size. For example, if the HIGHOFFLOAD value was specified as 80% and LOWOFFLOAD value was specified as 60%, 20% (80%-60%=20%) of the structure would be offloaded once offloading commenced.

The word “normally” has been used deliberately in the previous paragraphs. There are some situations when HIGHLOFFLOAD and LOWOFFLOAD parameters do not control the system logger offloading process.

When a coupling facility structure is defined, it is divided into two areas: One area holds *list elements*, and the other area holds *list entries*. List elements are units of logged data and are either 256 bytes or 512 bytes long. There is at least one element per log record. List entries are index pointers to the list elements. There is one list entry per log record.

Each log record places an entry in the list entry area of the structure, and the data is loaded as one or more elements in the list element area. **If the list entry area exceeds 90% of its capacity, all log streams are offloaded to DASD.** DASD offloading commences at this point, regardless of the current utilization of the log stream, and continues until an amount of data equal to the difference between the HIGHOFFLOAD threshold and the LOWOFFLOAD threshold for the log stream has been offloaded.

[1] The controls apply **only** to staging data set usage with DASD-only log streams. With coupling facility log streams, the controls apply to both coupling facility structure usage and staging data set usage if the log stream is duplexed to staging data sets.

This situation can occur when log steams share a structure, one log stream is used by an application issuing very few journal write requests, and other applications issue frequent journal write requests to log streams in the same structure. All log streams may be offloaded to DASD because of the frequent journal write requests by the other applications.

The primary disadvantage of encountering this situation is that the application that is infrequently writing to the log stream might not have its LOWOFFLOAD and HIGHOFFLOAD thresholds controlling the offload process. This can result in unpredictable offloading, and possibly undesirable performance.

For example, Log Stream A might have a HIGHOFFLOAD threshold of 80% and a LOWOFFLOAD threshold of 60%. Because of log stream activity by other applications writing to other log streams, the list entry area may exceed 90% of its capacity even though Log Stream A might be only 50% utilized. Although Log Stream A had not reached its HIGHOFFLOAD threshold, or even its LOWOFFLOAD threshold, data would be offloaded until 20% of the log stream was offloaded. This is the difference between 80% and 60%. After the offloading operation has completed, log stream A is at 30% utilization (50% minus 20%).

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file. CPExpert examines the SMF88STN variable (the structure name) to select information that applies only to coupling facility structures[2].

For these records, CPExpert examines the SMF88EFS variable (offloads for all log streams connected from this system to this structure because structure was 90% full) in the SMF Type 88 records. CPExpert produces Rule WLM703 when the SMF88EFS value exceeds the **STFULL90** guidance variable in USOURCE(CICGUIDE). The default value for the **STFULL90** is zero. Any non-zero value in the SMF88EFS variable indicates that the entry to element ratio is too high for the structure.

This problem occurs primarily when more than one log stream uses a coupling facility structure and the applications using the log streams write a significantly different rates. Consequently, the offloads are being triggered by all the entries being used rather than triggered by the HIGHOFFLOAD value.

Suggestion: When Rule WLM703 is produced, you should consider the following alternatives:

[2]The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

- Review the log streams that share the coupling facility structure. IBM recommends that log streams sharing a coupling facility structure have similar rates of writing and similar amounts of data written.

- Review the size of the list structure in the coupling facility, to determine whether the structure size should be increased.

- You can alter CPExpert's analysis by modifying the STFULL90 guidance variable in USOURCE(WLMGUIDE).

Reference: OS/390 MVS: Setting up a Sysplex

OS/390 (V2R2):	Section 9.4.3
OS/390 (V2R3):	Section 9.4.3
OS/390 (V2R4):	Section 9.4.3
OS/390 (V2R5):	Section 9.4.3
OS/390 (V2R6):	Section 9.4.3
OS/390 (V2R7):	Section 9.4.3
OS/390 (V2R8):	Section 9.4.3
OS/390 (V2R9):	Section 9.4.3
OS/390 (V2R10):	Section 9.4.3
z/OS (V1R1):	Section 9.4.3
z/OS (V1R2):	Section 9.4.3
z/OS (V1R3):	Section 9.4.3
z/OS (V1R4):	Section 9.4.3

©Copyright 1995, Computer Management Sciences, Inc.

Rule WLM704: Interim storage was not efficiently used for log stream

Finding: The SMF Type 88 data showed that Interim storage (the coupling facility structure for the log stream) was not efficiently used for the log stream.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[1], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only[2] log streams. The main difference between the two types of log streams is the storage medium that the system logger uses to hold interim log data:

- With a coupling facility log stream, interim storage is contained in coupling facility list structures. The system logger duplexes the log stream to either (1) MVS data space areas associated with the system logger address space or (2) staging data sets, depending on whether the coupling facility is failure-independent.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with

[1] Interim storage is sometimes referred to as "primary" storage.

[2] DASD-only log streams are supported beginning with OS/390 Version 2 Release 4.

©Copyright 1995, Computer Management Sciences, Inc.

the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to staging data sets

Interim storage normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled.

For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure[3] is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the coupling facility to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading log stream data from the coupling facility list structure to DASD log data sets when 80% or more of the structure has been used.

- The **low offload threshold** is the point in the coupling facility structure, as a percent of space consumed, where the system logger stops offloading coupling facility log data to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

Once log stream data has been offloaded, the MVS system logger releases the storage in the list structure, so the space in the structure can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

[3] Please note that under certain conditions, a coupling facility log stream might be duplexed to staging data sets. If this should be the case, the HIGHOFFLOAD value applies to the staging data sets as well as to the coupling facility structure. See Rule WLM706 for additional information. For DASD-only log streams, duplexing to staging data sets is automatic.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert computes the percent of ineffective use of interim storage (PCTINTST) by applying the following algorithm:

$$PCTINTST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where:

SMF88SAB	=	Bytes deleted after being offloaded
SMF88SIB	=	Bytes deleted before being offloaded

CPExpert compares the computed PCTINST with the **PCTINST** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM704 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTINST** guidance variable.

The default value for the **PCTINST** guidance variable is zero, indicating that CPExpert should produce Rule WLM704 whenever interim storage use was not effective.

Suggestion: The delete after offload percent is a key indicator that log tail deletion is not working as effectively as it should. If significant values appear in this percent, you should consider the following alternatives:

- Consider increasing the HIGHOFFLOAD threshold value.

- For CICS/TS, verify that SYSLOG=KEEP is not specified as a System Initialization Table (SIT) parameter (this suggestion applies only to CICS/TS Release 1.1, as the SYSLOG keyword was made obsolete with CICS/TS Release 1.2). The SYSLOG=KEEP option inhibits CICS from deleting data from the system log, even though the data is no longer needed. IBM strongly recommends that the SYSLOG=NOKEEP option be used, and the SYSLOG keyword was removed from the SIT with CICS/TS Release 1.2.

- Verify that there are not any long running transactions making recoverable updates without syncpointing.

- For CICS/TS, examine the System Initialization Table (SIT) values for this region, and determine whether AKPFREQ is zero or is too high. With a CICS/ESA 4.1 region (or earlier), the AKPFREQ parameter

specifies the number of consecutive blocks written to the system log data set. However, with CICS/TS for OS/390, the AKPFREQ parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken.

- If AKPFREQ=0, CICS cannot perform log tail deletion until shutdown, by which time the system log will have spilled to secondary storage. This situation would elongate shutdown and cause unnecessary overhead.

- The AKPFREQ parameter has a significant impact on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. During AKP processing, the system logger deletes records that are no longer of interest to CICS and moves records to DFHSHUNT for those tasks that did write any log records within the last AKP interval.

- In an MRO environment, the keypoint program uses an appreciable amount of CPU capacity in processing persisting units of work such as those relating to mirror transactions waiting to process an implicit forget. This is exacerbated when the AKPFREQ value is low. An optimum setting of AKPFREQ allows many of these persistent units of work to complete during normal transaction processing activity. This minimizes the CPU processing used by the keypoint program. IBM suggests that you exercise caution in reducing the value of AKPFREQ below the default value.

- Consider increasing the size of the coupling facility structure.

- You can alter CPExpert's analysis by modifying the **PCTINTST** guidance variable in USOURCE(WLMGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*:
Section 4.6.1 (Monitoring the logger environment).
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.2 Performance Guide:
Section 4.6.2: Monitoring the logger environment.
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.3 Performance Guide:
Section 4.8.2: Monitoring the logger environment.
Section 4.8.7: Activity keypoint frequency (AKPFREQ).

©Copyright 1995, Computer Management Sciences, Inc.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

Rule WLM705: Staging data sets not efficiently used, DASD-only log stream

Finding: The SMF Type 88 data showed that staging data sets were not efficiently used for a DASD-only[1] log stream.

Impact: This finding has a LOW IMPACT or MEDIUM IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*[2], where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium that the system logger uses to hold interim log data:

- With a coupling facility log stream, interim storage is contained in coupling facility list structures. The system logger duplexes the log stream to either (1) MVS data space areas associated with the system logger address space or (2) staging data sets, depending on whether the coupling facility is failure-independent.

- With a DASD-only log stream, interim storage is contained in local storage buffers on the system (as MVS data space areas associated with

[1] DASD-only log streams are supported beginning with OS/390 Version 2 Release 4.

[2] Interim storage is sometimes referred to as "primary" storage.

©Copyright 1995, Computer Management Sciences, Inc.

the system logger address space). With a DASD-only log stream the system logger duplexes the log stream to staging data sets

Interim storage normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled. For log streams defined in coupling facility list structures, the parameters apply to the coupling facility structures[3].

For log streams defined as DASD-only, these parameters apply to the log stream staging data set, as follows:

- When the staging data set is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the staging data set to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading log stream data from the staging data set to DASD log data sets when 80% or more of the staging data set has been used.

- The **low offload threshold** is the point in the staging data set, as a percent of space consumed, where the system logger stops offloading log data in the staging data set to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

Once log stream data has been offloaded, the MVS system logger releases the storage in the staging data sets, so the space in the staging data sets can be used to hold new log blocks. From an application point of view, the actual location of the log data in the log stream is transparent.

Applications using system logger services (such as CICS/Transaction Server for OS/390) often manage the system log by deleting records for completed units of work during activity keypoint processing (this is also called log-tail deletion). The number of bytes deleted from the system log after writing to offload data sets should be very low. Unnecessary overhead is incurred when data is moved to the offload data sets, only to be later deleted. With an appropriately sized log stream, the system log data remains in interim storage, and the overhead of data spilling to DASD simply to be deleted later is avoided.

[3] The parameters will also apply to staging data sets if the log stream is duplexed to staging data sets. Problems with staging data set threshold being encountered are analyzed in Rule WLM705.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert computes the percent of ineffective use of staging data sets (PCTLOCST) by applying the following algorithm to DASD-only log streams:

$$PCTLOCST = \frac{SMF88SAB}{SMF88SIB + SMF88SAB}$$

where

SMF88SAB = Bytes deleted after being offloaded

SMF88SIB = Bytes deleted before being offloaded

CPExpert compares the computed PCTLOCST with the **PCTLOCST** guidance variable in USOURCE(WLMGUIDE). CPExpert produces Rule WLM704 when the percent ineffective use of use of interim storage exceeds the value specified by the **PCTLOCST** guidance variable.

The default value for the **PCTLOCST** guidance variable is 0, indicating that CPExpert should produce Rule WLM705 whenever DASD staging data set use was not effective.

Suggestion: The delete after offload percent is a key indicator that log tail deletion is not working as effectively as it should. If significant values appear in this percent, you should consider the following alternatives:

- For CICS/TS, verify that SYSLOG=KEEP is not specified as a System Initialization Table (SIT) parameter (this suggestion applies only to CICS/TS Release 1.1, as the SYSLOG keyword was made obsolete with CICS/TS Release 1.2). The SYSLOG=KEEP option inhibits CICS from deleting data from the system log, even though the data is no longer needed. IBM strongly recommends that the SYSLOG=NOKEEP option be used, and the SYSLOG keyword was removed from the SIT with CICS/TS Release 1.2.

- Verify that there are not any long running transactions making recoverable updates without syncpointing

- Consider increasing the HIGHOFFLOAD threshold value.

- For CICS/TS, examine the System Initialization Table (SIT) values for this region, and determine whether AKPFREQ is zero or is too high. With a CICS/ESA 4.1 region (or earlier), the AKPFREQ parameter

specifies the number of consecutive blocks written to the system log data set. However, with CICS/TS for OS/390, the AKPFREQ parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken.

- If AKPFREQ=0, CICS cannot perform log tail deletion until shutdown, by which time the system log will have spilled to secondary storage. This situation would elongate shutdown and cause unnecessary overhead.

- The AKPFREQ parameter has a significant impact on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. During AKP processing, the system logger deletes records that are no longer of interest to CICS and moves records to DFHSHUNT for those tasks that did write any log records within the last AKP interval.

- In an MRO environment, the keypoint program uses an appreciable amount of CPU capacity in processing persisting units of work such as those relating to mirror transactions waiting to process an implicit forget. This is exacerbated when the AKPFREQ value is low. An optimum setting of AKPFREQ allows many of these persistent units of work to complete during normal transaction processing activity. This minimizes the CPU processing used by the keypoint program. IBM suggests that you exercise caution in reducing the value of AKPFREQ below the default value.

- Consider increasing the size of the DASD staging data sets.

- You can alter CPExpert's analysis by modifying the **PCTLOCST** guidance variable in USOURCE(WLMGUIDE).

Reference: *CICS/TS Release 1.1 Performance Guide*:
Section 4.6.1 (Monitoring the logger environment).
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.2 Performance Guide:
Section 4.6.2: Monitoring the logger environment.
Section 4.6.7: Activity keypoint frequency (AKPFREQ).

CICS/TS Release 1.3 Performance Guide:
Section 4.8.2: Monitoring the logger environment.
Section 4.8.7: Activity keypoint frequency (AKPFREQ).

©Copyright 1995, Computer Management Sciences, Inc.

CICS/TS for z/OS Release 2.1 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

CICS/TS for z/OS Release 2.2 *Performance Guide*: Chapter 22:
Monitoring the logger environment.
Activity keypoint frequency (AKPFREQ).

Rule WLM706: DASD staging data set high threshold was reached

Finding: The SMF Type 88 data showed that the DASD staging data set high threshold was reached.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data. The finding applies only to log streams that are defined to use a coupling facility.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is “hardened” for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is coupling facility list structures.
- In a DASD-only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Additionally, for data integrity there exists duplexed storage, so that if one system or component fails, the log stream can be recovered from the duplexed storage. These concepts differ, depending on whether the log stream is defined for a coupling facility or for DASD-only.

©Copyright 1995, Computer Management Sciences, Inc.

- If the primary storage is defined as a list structure in a coupling facility, the duplexed data can be retained in another coupling facility, or can be retained in *staging data sets*. Staging data sets are used when the coupling facility is in the same CPC, or uses volatile storage.

- If the primary storage is defined as DASD-only, the duplexed data is retained in *staging data sets*.

When a log stream in a coupling facility is duplexed to staging data sets, the system logger automatically makes a duplicate copy of the data every time data is written to a log stream. This is done to protect against data loss due to coupling facility problems or due to system failure. The duplicate copy is kept in the staging data sets until the data is off-loaded from the coupling facility structure to DASD log data sets. After the data is off-loaded to DASD log data sets, the system logger discards the duplicate copy of the log data.

Interim storage in a coupling facility structure normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage being filled. For log streams defined in coupling facility list structures, these parameters apply as follows:

- When the coupling facility structure is filled to the **high offload threshold** point or beyond, the system logger begins offloading data from the coupling facility to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **low offload threshold** is the point in the coupling facility structure, as a percent of space consumed, where the system logger stops offloading coupling facility log data to log stream DASD data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

From the above description, the amount of data that normally is offloaded is the difference between HIGHOFFLOAD and LOWOFFLOAD, as percentages of the coupling facility list structure size. For example, if the HIGHOFFLOAD value was specified as 80% and LOWOFFLOAD value was specified as 60%, 20% (80%-60%=20%) of the structure would be offloaded once offloading commenced.

For log streams in a coupling facility that are duplexed to staging data sets, the values of the HIGHOFFLOAD and LOWOFFLOAD parameters **apply**

to the staging data sets as well as to the coupling facility structure. This is simply because if the staging data sets become full, MVS would not be able to continue duplexing data and there would be a data integrity exposure in case of failure. Consequently, if a staging data set fills up **before** an offload of a log stream in a coupling facility structure is triggered by the high threshold specification, an offload will be triggered because of the full staging data set.

When a staging data set reaches the high threshold, the system logger immediately offloads data from the coupling facility to DASD log data sets, even if the coupling facility usage for the log stream is below the high threshold. Thus, if the staging data sets are small in comparison to the coupling facility structure size for a log stream, the staging data sets will keep filling up and the system logger will frequently offload coupling facility data to DASD log data sets. This means that your installation would experiences frequent (and unexpected) offloading overhead that could affect performance[1].

IBM's "Setting up a Sysplex" document (Section 9.4.5.6: Monitoring Staging Data Set Usage Log Streams) contains the following comments:

> "Whether your staging data sets are defined by system logger or on the STG_SIZE parameter, you should carefully monitor your staging data sets. This applies to both coupling facility and DASD-only log streams, and is important because the consequences of having your staging data set fill up can be quite disruptive. When a system's staging data set fills up, system logger applications on that system will not be able to write to the log stream until log data can be offloaded to DASD, which frees up space in the staging data set. Thus, when your staging data sets are too small, system logger will perform coupling facility offloading more frequently than the HIGHOFFLOAD and LOWOFFLOAD thresholds defined for the log stream would otherwise require. This can negatively affect the performance of all the log streams in that structure."

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88STN variable (the structure name) in the MXG TYPE88 data set to select records that apply only to coupling facility structures[2]. For these records, CPExpert examines the SMF88ETT variable (the number of times the system logger detected a Staging Data Set Threshold hit condition). CPExpert produces Rule WLM706 when the

[1] If your staging data sets are too small, you also run the risk of filling them up completely. If this occurs, system logger immediately begins offloading the coupling facility log data in DASD log data sets to harden it. System logger applications will be unable to log data until system logger can free up staging data set space. This serious situation is evaluated by Rule WLM702.

[2] The SMF88STN variable will be *DASDONLY* for log streams that are DASD-only log streams.

SMF88ETT value exceeds the **STDSHIGH** guidance variable in USOURCE(WLMGUIDE).

The default value for the **STDSHIGH** is 0, indicating that CPExpert should produce Rule WLM706 whenever a Staging Data Set Threshold was encountered during an RMF interval.

Suggestion: IBM suggests that you size the staging data sets larger than the coupling facility structure size for the log streams.

While you can modify CPExpert's analysis by altering the STDSHIGH guidance variable, you should not do so unless you have unusual circumstances.

Reference: OS/390 Setting up a Sysplex

OS/390 (V2R2):	Section 9.2.4
OS/390 (V2R3):	Section 9.2.5
OS/390 (V2R4):	Section 9.2.6
OS/390 (V2R5):	Section 9.2.6
OS/390 (V2R6):	Section 9.2.6
OS/390 (V2R7):	Section 9.2.6
OS/390 (V2R8):	Section 9.2.6
OS/390 (V2R9):	Section 9.2.6
OS/390 (V2R10):	Section 9.2.6
z/OS (V1R1):	Section 9.2.6
z/OS (V1R2):	Section 9.4.6
z/OS (V1R3):	Section 9.4.6
z/OS (V1R4):	Section 9.4.6

Rule WLM707: Frequent log stream DASD-shifts occurred

Finding: The SMF Type 88 data showed that frequent log stream DASD-shifts occurred.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is coupling facility list structures.

- In a DASD-only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

A log stream can have data in multiple DASD log data sets. As an offload data set becomes full, the system logger automatically allocates a new one for the log stream. This process is known as a "DASD-shift" and *generates considerable overhead*. Consequently, a "DASD-shift" should not occur frequently. IBM suggests that "DASD-shifts" should occur no more than once per hour.

©Copyright 1995, Computer Management Sciences, Inc.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88EDS variable (the number of log stream DASD shifts during the SMF interval). Recall that IBM suggests that you not have more than one DASD shift per hour. However, an SMF recording interval typically is less than an hour (normally the interval is 15 minutes). Consequently, CPExpert calculates the number of SMF intervals in an hour and tracks the number of DASD shifts that occur during any hour.

CPExpert produces Rule WLM707 when the number of DASD shifts that occur during any hour exceeds the **LGSHIFTS** guidance variable in USOURCE(WLMGUIDE).

The default value for the **LGSHIFTS** is one, indicating that CPExpert should produce Rule WLM707 when more than one log stream DASD shift occurred during any hour.

Suggestion: If CPExpert produces Rule WLM707, you should consider the following alternatives:

- If more than one DASD shift occurs per hour, you should increase the size of the offload data sets. IBM recommends that you size the offload data sets as large as your installation can afford to make them. This will minimize the number of log data sets required to represent a log stream. It will also minimize the number of times that system logger must reallocate and switch to using a new log data set when an old one becomes full.

- You can alter CPExpert's analysis by changing the value of the **LGSHIFTS** guidance variable in USOURCE(WLMGUIDE).

Reference: OS/390 Setting up a Sysplex

OS/390 (V2R4):	Section 9.4.5
OS/390 (V2R5):	Section 9.4.5
OS/390 (V2R6):	Section 9.4.5
OS/390 (V2R7):	Section 9.4.5
OS/390 (V2R8):	Section 9.4.5
OS/390 (V2R9):	Section 9.4.5
OS/390 (V2R10):	Section 9.4.5
z/OS (V1R1):	Section 9.4.5
z/OS (V1R2):	Section 9.4.5
z/OS (V1R3):	Section 9.4.5
z/OS (V1R4):	Section 9.4.5

Rule WLM708: Log stream caused structure to reach high threshold

Finding: The SMF Type 88 data showed that the log stream caused its coupling facility structure to reach high threshold.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications.

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is "hardened" for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is contained in coupling facility list structures.

- In a DASD-only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Interim storage normally is "offloaded" to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and

©Copyright 1995, Computer Management Sciences, Inc.

LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage[1] being filled.

- When the interim storage (either coupling facility structure or staging data set) is filled to the **HIGHOFFLOAD threshold** point or beyond, the system logger begins offloading log data to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **LOWOFFLOAD threshold** is the point in the interim storage (coupling facility structure or staging data set), as a percent of space consumed, where the system logger stops offloading log data to DASD log data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility structure to DASD log data sets or if the system logger configuration is tuned incorrectly. The Type-3 completions are analyzed by Rule MVS309.

[1] The controls apply **only** to staging data set usage with DASD-only log streams. With coupling facility log streams, the controls apply to both coupling facility structure usage and staging data set usage if the log stream is duplexed to staging data sets.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88SC2 variable (Count of Type-2 completions) in the SMF Type 88 records. CPExpert produces Rule WLM708 when the SMF88SC2 value exceeds the **STRC2** guidance variable in USOURCE(WLMGUIDE). The default value for the **STRC2** is zero, indicating that CPExpert should produce Rule WLM708 whenever the HIGHOFFLOAD threshold was reached in an SMF interval.

Suggestion: The number of Type-2 completions is simply a count of the number of times the HIGHOFFLOAD threshold for the coupling facility structure was reached based on writes to the specific log stream. Reaching the HIGHOFFLOAD threshold might or might not be an indication of a problem.

- You might wish log data to be frequently "hardened" to a DASD log data set. In this situation, you would define a relatively small coupling facility structure or specify a relatively low value for the HIGHOFFLOAD threshold. Consequently, you would expect to have Type-2 completions relatively often and a relatively large number of Type-2 completions would not be a cause for concern.

  If this condition applies to the log stream, you should consider "turning off" Rule WLM708 for this log stream. Please refer to Section 3 for instructions on how to "turn off" rules and for instructions on how to specify guidance for individual log streams or structures.

- You might have multiple log streams sharing the coupling facility structure, or you might not wish to experience the overhead of offloading. In this situation, a large number of Type-2 completions (with the corresponding overhead of offloading) might be cause for alarm.

  If this condition applies to the structure, you should consider separating the log streams that use the structure (either creating a new coupling facility structure or using a different distribution scheme for the log streams amongst the structures that are defined. As a general guidance, you should not have log streams with different characteristics sharing the same coupling facility structure.

Reference: OS/390 MVS System Management Facilities
OS/390 (V2R4): Section 9.1.1.2
OS/390 (V2R5): Section 9.1.1.2
OS/390 (V2R6): Section 9.1.1.2
OS/390 (V2R7): Section 9.1.1.2

OS/390 (V2R8):	Section 9.1.1.2
OS/390 (V2R9):	Section 9.1.1.2
OS/390 (V2R10):	Section 9.1.1.2
z/OS (V1R1):	Section 9.1.1.2
z/OS (V1R2):	Section 9.1.1.2
z/OS (V1R3):	Section 9.1.1.2
z/OS (V1R4):	Section 9.1.1.2

Rule WLM709: Log stream consumed most of structure resources

Finding: The SMF Type 88 data showed that the log stream consumed most of its coupling facility structure resources.

Impact: This finding has a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of your computer system. The level of impact depends on the applications using the log stream, and the extent to which log stream delays effects the performance of these applications..

Logic flow: This is a basic finding, based on an analysis of the SMF Type 88 system logger data.

Discussion: The system logger is an MVS component that allows an application to log data from a sysplex. You can log data from one system or from multiple systems across the sysplex.

Please refer to Rule WLM701 for more general information about the MVS system logger.

Data in a log stream is contained in two kinds of storage: (1) *interim storage*, where data can be accessed quickly without incurring DASD I/O, and (2) *DASD log data set storage*, where data is “hardened” for longer term access. When the interim storage medium for a log stream reaches a user-defined threshold, the log data is offloaded to DASD log data sets.

There are two types of log streams: coupling facility log streams and DASD-only log streams. The main difference between the two types of log streams is the storage medium system logger uses to hold interim log data:

- In a coupling facility log stream, interim storage is contained in coupling facility list structures.

- In a DASD-only log stream, interim storage is contained in local storage buffers on the system, as an MVS data space areas associated with the system logger address space.

Interim storage normally is “offloaded” to DASD log data sets based on two parameters associated with each log stream: the HIGHOFFLOAD and

©Copyright 1995, Computer Management Sciences, Inc.

LOWOFFLOAD parameters. The values for these parameters are expressed as a percent of the interim storage[1] being filled.

- When the interim storage (either coupling facility structure or staging data set) is filled to the **HIGHOFFLOAD threshold** point or beyond, the system logger begins offloading log data to the DASD log stream data sets. For example, if the HIGHOFFLOAD parameter is specified as 80% (this is the default value), the system logger normally would begin offloading interim storage to DASD log data sets when 80% or more of the structure is used.

- The **LOWOFFLOAD threshold** is the point in the interim storage (coupling facility structure or staging data set), as a percent of space consumed, where the system logger stops offloading log data to DASD log data sets. The default LOWOFFLOAD parameter value is 0%, indicating that the system logger will offload all the log stream to DASD log data sets once offloading has commenced.

When a system logger user issues the IXGWRITE macro for a coupling facility log stream, the system logger writes to the coupling facility structure. When the write completes, the system logger categorizes the event as a *Type-1*, *Type-2*, or *Type-3* completion. The categorization indicates how much space in the structure is being used by the log stream when the completion occurred.

- A *Type-1* completion indicates that, after the write completed, the percentage of the structure space used was less than the HIGHOFFLOAD threshold, meaning that system logger is using the coupling facility successfully. This is a desired completion status.

- A *Type-2* completion indicates that, after the write completed, the percentage of the structure space used was equal to or greater than the HIGHOFFLOAD threshold. This means that the system logger begins managing storage resources by migrating data from the coupling facility to DASD log data sets.

  The number of Type-2 completions is simply a count of the number of times the HIGHOFFLOAD threshold for the coupling facility structure was reached based on writes to the specific log stream. Reaching the HIGHOFFLOAD threshold might or might not be an indication of a problem.

  - You might wish log data to be frequently “hardened” to a DASD log data set. In this situation, you would define a relatively small coupling

[1] The controls apply **only** to staging data set usage with DASD-only log streams. With coupling facility log streams, the controls apply to both coupling facility structure usage and staging data set usage if the log stream is duplexed to staging data sets.

facility structure or specify a relatively low value for the HIGHOFFLOAD threshold. Consequently, you would expect to have Type-2 completions relatively often and a relatively large number of Type-2 completions would not be a cause for concern.

- You might have multiple log streams sharing the coupling facility structure, or you might not wish to experience the overhead of offloading. In this situation, a large number of Type-2 completions (with the corresponding overhead of offloading) might be cause for alarm.

- A *Type-3* completion indicates that a given log stream is close to consuming all the space in the coupling facility. A Type-3 completion can occur if there is a failure which prevents the system logger from promptly moving data from the coupling facility structure to DASD log data sets or if the system logger configuration is tuned incorrectly.

  For example, the system logger's access to its DASD log data sets would be slowed if those data sets reside on the same device as some other heavily-used data sets.

  A Type-3 can also occur if many log streams are defined to share the same structure, because each newly defined log stream causes the system logger to dynamically repartition storage among the existing log streams.

  If a log stream has a large proportion of Type-3 completions, the system logger is getting dangerously close to the STRUCTURE FULL condition.

The MVS system logger writes SMF Type 88 records containing statistics for each connected log stream. This information is available as MXG TYPE88 file.

CPExpert examines the SMF88SC3 variable (Count of Type-3 completions) in the SMF Type 88 records. CPExpert produces Rule WLM708 when the SMF88SC3 value exceeds the **STRC3** guidance variable in USOURCE(WLMGUIDE). The default value for the **STRC3** is zero, indicating that CPExpert should produce Rule WLM709 whenever the space used by a log caused the coupling facility structure to reach a critical amount.

Suggestion: If this finding is produced, determine whether there was a failure that caused the system logger to be unable to promptly offload data. If a failure did occur, you probably should ignore this finding. If a failure was not experienced, you should consider the following alternatives:

- Determine whether the system logger configuration is tuned incorrectly. The system logger might be unable to offload data promptly if the DASD log data sets experience I/O contention with other systems data sets.

- Review the structure size, to ensure that the structure is adequately sized for the log stream(s) using the structure.

- Review the number of log streams assigned to the coupling facility structure. The system logger might not be able to respond adequately if too many log streams are defined to share the same structure.

- Examine the application responsible for the log stream activity to determine whether its use of the log stream has increased, and whether this increase is expected.

- Review the HIGHOFFLOAD and LOWOFFLOAD parameters for the log stream to determine whether these should be adjusted. If either parameter value is too large, the system logger might not be able to respond adequately. The system logger might not have time to offload sufficient log stream data when the HIGHOFFLOAD parameter value is reached, before the log stream uses most of the structure. The system logger might offload only a relatively small amount of data once offloading commences, if the LOWOFFLOAD parameter is too high. Either of these situations could indicate that the parameters are too large, or could simply be the result of the coupling facility structure being too small.

- Review the size of the off-load data sets. These should be large enough to avoid too many "DASD shifts"--that is, new data set allocations. Rule WLM707 would be produced by CPExpert if too many DASD shifts occurred. However, you might have altered the guidance to CPExpert for Rule WLM707. In this case, Rule WLM707 might not be produced even though DASD shifts could have delayed the offloading of the log stream(s) assigned to the coupling facility structure.

Reference: OS/390 MVS System Management Facilities

OS/390 (V2R6):	Section 9.1.1.2
OS/390 (V2R7):	Section 9.1.1.2
OS/390 (V2R8):	Section 9.1.1.2
OS/390 (V2R9):	Section 9.1.1.2
OS/390 (V2R10):	Section 9.1.1.2
z/OS (V1R1):	Section 9.1.1.2
z/OS (V1R2):	Section 9.1.1.2
z/OS (V1R3):	Section 9.1.1.2
z/OS (V1R4):	Section 9.1.1.2

Your turn:

This manual has described how to use the WLM Component to analyze performance constraints with IBM's Workload Manager.

We would appreciate receiving any comments you have regarding this document (style, content, clarity, etc.), or suggestions for improving the WLM Component (ease of use, new rules, changes to rules, etc.). Please send your comments to:

Don Deese
Computer Management Sciences, Inc.
6076-D Franconia Road
Alexandria, VA 22310
www.cpexpert.com

Comments: