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.