Rule WLM008: DUR value may be too large for TSO Period 1

Finding: CPExpert believes that the DUR value may be too large for a service class that CPExpert believes describes TSO Period 1 transactions.

Impact: This finding has a MEDIUM impact or HIGH impact on the performance of TSO interactive transactions.

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

Discussion: The Workload Manager allows installations to group similar work into service classes. For example, TSO transactions might be placed into one service class, batch jobs might be placed into another service class, etc. Users define service classes by specifying the name of the service class and specifying the performance goal, service class, and service class importance.

An installation may wish to assign different performance attributes to work depending upon the inherent resource demands of the work. For example, all TSO transactions might be assigned to a particular service class. However, installation management might wish to assign a high performance goal and goal importance to trivial TSO transactions, non-trivial (but still quite interactive) TSO transactions might be assigned a slightly lower performance goal and goal importance, and relatively non-interactive TSO transactions might be assigned still lower performance goal and goal importance. This differentiation among TSO transactions **based on the inherent characteristics of the transactions** is accomplished via the "service class period" option.

A service class may be broken into multiple service class **periods**. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional service class periods (although typically no more than two or three additional service class periods are defined). Each service class period can have its own performance goals, defined to the Workload Manager via the Workload Manager ISPF panels.

An address space (TSO transaction, batch job step, etc.) begins in Period 1 of the service class to which it is assigned. The address space transitions from Period 1 to Period 2 (and to subsequent periods), based upon the accumulation of "service" by the transaction. The "service" required by the address space is a combination of CPU resources, I/O resources, and memory resources. The actual resources used by the address space are adjusted by the CPU, IOC, SRB, and MSO service coefficients to yield the CPU, I/O, and memory "service" requirements of the address space.

When the address space accumulates more than a specified amount of service, the address space will be reassigned to the next lower service class period (if multiple periods are defined)[1]. The amount of service controlling when an address space is reassigned to the next lower service class period is specified by the value of the **DUR** keyword in the service definition. The DUR keyword is associated with any service class period which has subsequent service class periods.

The normal purpose of defining multiple service class periods is to give higher priority to interactive transactions, short batch job steps, etc. Overall response is decreased (and overall throughput is increased) when address spaces requiring relatively few resources are processed at a higher priority than those address spaces requiring substantial resources.

It is instructive to examine the resources that may be in conflict. The resources mainly consist of CPU cycles, processor storage, and I/O operations.

- Prior to MVS/ESA SP5, the CPU resource was the main area of conflict. CPU dispatching priority is established at the performance group period level. An address space in a particular performance group period executed at the same CPU dispatching priority as all other address spaces in the performance group period[2]. Non-interactive transactions could cause interactive transactions to be denied access to a processor for an unacceptable interval.

- Conflict with processor storage occurs when non-interactive transactions require either central storage or expanded storage, causing central storage page fault or expanded storage delays for interactive transactions. Non-interactive transactions often (but not always) require a significant amount of processor storage.

---

[1]Note that the address space is not reassigned immediately upon accumulation of the specified amount of service. The reassignment is performed only when the System Resources Manager (SRM) evaluates the address space for changed conditions. Therefore, the SRM cannot determine when the **exact** amount of service specified in the DUR parameter is used; the SRM can only determine when the DUR value is exceeded.

The SRM checks for transactions exceeding their DUR value only when the SRM gains control. Each address space normally is evaluated every three SRM seconds, and reassignment is performed only if the address space has accumulated more than the value specified for the DUR keyword. Consequently, an address space will always accumulate **more** service than specified by the DUR keyword before being reassigned.

[2]This discussion ignores the implications of time-slicing algorithms and ignores the implications of the Mean-time-to-Wait (MTTW) dispatching algorithms. These special case algorithms are not normally used for Period 1 or Period 2 of multi-period performance groups.

- Non-interactive transactions can cause conflict with I/O operations by either path, controller, or device delays to interactive transactions. Fortunately, (1) modern systems normally have multiple paths to devices and path delay is not a serious concern, (2) and non-interactive transactions normally do not access the same devices as interactive transactions, and (3) interactive transactions normally do not execute many I/O operations. The combined effect is that conflict for I/O operations rarely is a cause of performance problems between interactive and non-interactive transactions.

As the above discussion illustrates, the main concern prior to MVS/ESA SP5 was CPU dispatching. This concern has been almost completely eliminated with SP5.

- Prior to SP5, CPU dispatching was on a "first come first served" or FIFO basis from the Dispatcher's True Ready Queue. A non-interactive transaction could easily deny an interactive transaction from access to a processor for a prolonged period; the non-interactive transaction could retain control of a processor until it either completed or it voluntarily entered a Wait state.

- With SP5, the Dispatcher algorithm has been redesigned. One aspect of the redesign is that each interrupted dispatchable unit (a TCB or SRB) is placed at the end of the dispatching queue for its current dispatching priority.

  For example, suppose that a dispatchable unit was executing with a priority of 240. When the dispatchable unit was interrupted, it would be placed behind all other dispatchable units on the dispatch queue with a dispatching priority of 240 (but ahead of all dispatchable units with a dispatching priority of 239 and lower).

  Since interrupts occur frequently, there is a constant adjustment of dispatching order. This readjustment requires practically no additional overhead, as only queue pointers are being manipulated.

  One effect of the redesign is to prevent any dispatchable unit from "seizing" a processor and denying access to other dispatchable units at the same priority. Since a dispatchable unit cannot deny access to other dispatchable units, a non-interactive transaction cannot deny CPU access to a Ready interactive transaction.

Consequently, with SP5, there is less need to "separate" non-interactive and interactive transactions, and there is less need to specify multiple periods. Nonetheless, the need to separate interactive and non-interactive transactions still exists.

The SRM will be able to differentiate between interactive and non-interactive transactions only if the values specified for the DUR keyword roughly correspond to the resource requirements of trivial, interactive, and non-interactive transactions.

If the value specified for the DUR keyword for Period 1 is too large, non-trivial and non-interactive transactions will execute with the same performance management controls as those given to trivial transactions. This would defeat the purpose of breaking the service class into multiple service class periods.

CPExpert scans the Service Class Description (SMF Type 72 field R723MCDE) for the word "TSO" and assumes that the service class describes TSO transactions if "TSO" is encountered.

If CPExpert identifies a service class describing TSO transactions, CPExpert examines the DUR value and the service class period. If a DUR value is specified and the period is Period 1, CPExpert analyzes the DUR value[3]. CPExpert concludes that the DUR value is too large for TSO Period 1 transactions if the DUR value is greater than 100 (for service policies which have MSO less than 1) or greater than 300 (for service policies which have MSO 1 or higher).

This conclusion is based upon comparing the DUR value with the DUR values specified by other installations. After making this comparison, CPExpert found that the DUR value specified for TSO Period 1 was much higher than the value commonly specified for trivial TSO transactions.

The following example illustrates the output from Rule WLM008:

```
RULE WLM008:  THE DUR VALUE MAY BE TOO LARGE FOR TSO PERIOD 1.

   CPExpert noticed that TSOUSERS Service Class (Period 1) had the word
   "TSO" in its Service Class Description.  Consequently CPExpert assumes
   that the service class consists of TSO transactions.  The DUR value for
   the TSOUSERS Service Class was specified as 400.  CPExpert believes that
   the DUR value may be too large for TSO Period 1 transactions.  TSO
   Period 1 normally should handle interactive TSO transactions.  These
   interactive transactions typically use few system resources.  If the
   DUR value is too large for TSO Period 1, non-interactive transactions
   will remain in TSO Period 1 for a longer time before they are migrated
   to TSO Period 2.  During the time that the non-interactive transactions
   remain in TSO Period 1, they interfere with interactive transactions.
   If the TSOUSERS Service Class does describe TSO transactions, you may wish
   to reduce the DUR value so that Period 1 of this service class handles
   interactive TSO transactions and other periods handle less interactive
   TSO transactions.
```

[3]This analysis can be overridden by the **MAXDUR** guidance variable in USOURCE(WLMGUIDE).

Suggestion: CPExpert suggests that you review the DUR value specified for TSO Period 1. Unless there are unique conditions at your installation, CPExpert suggests that the DUR value be significantly lowered.

Most installations have established the DUR value for TSO Period 1 such that about 80-90% of the TSO transactions end in Period 1. There is nothing "magic" about these percentages! They are simply general guidance that has been provided by IBM. If your situation warrants a different percentage, set the DUR value accordingly.

The proper specification for the DUR value for TSO Period 1 depends entirely upon the resource requirements of trivial TSO transactions at your installation.

Some installations operate quite efficiently with only about 50% of the TSO transactions ending in Period 1! In fact, many installations have established an extremely low DUR value for TSO Period 1 (e.g., DUR=50) with the intent of rapidly processing extremely trivial TSO transactions (for example, a TSO user simply pressing a PF key in ISPF). More common specifications are DUR=100 for TSO Period 1.

The default analysis CPExpert makes depends upon the setting of the MSO service coefficient.

- If MSO is set to less than 1, CPExpert checks the DUR value against 100.
- If MSO is set to 1 or more, CPExpert checks the DUR against 300 as central storage use will represent a significant part of the service accumulation.

You can alter the analysis done by CPExpert by using the **MAXDUR** guidance variable in USOURCE(WLMGUIDE) to provide CPExpert with guidance about what DUR value you consider reasonable.

Reference: OS/390 (V1R3) MVS Planning: Workload Management
Chapter 8.2: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R4) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R5) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R6) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R7) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R8) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R9) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

OS/390 (V2R10) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R1) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R2) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R3) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients

z/OS (V1R4) MVS Planning: Workload Management
Section 8.3: Using Performance Periods
Chapter 11: Defining Service Units and Coefficients