

ANALYZING RMF SERVICE CLASS DATA AND WORKLOAD ACTIVITY REPORTS

Donald R. Deese

©Copyright 1998, Computer Management Sciences, Inc.

A new Workload Activity Report was introduced with MVS (Goal Mode) to show the performance of service classes executing under the Workload Manager. This paper describes each important field in the RMF Workload Activity Report and identifies the SMF data elements used in the field. Additionally, the paper explains why impossibly high values sometimes appear in the Workload Activity Report for subsystem transaction service class data.

1.0 INTRODUCTION

In earlier versions of MVS, installations classified work using the IEAICSxx member of SYS1.PARMLIB. After applying classification rules in IEAICSxx, the resulting work would be assigned to performance groups.

With MVS (Goal Mode), the same process conceptually applies. However, instead of assigning work to performance groups, work is assigned to a service class (or to a report class) after applying classification rules.

A service class can represent any collection of workloads that can be classified using the workload classification schemes. Users specify a performance goal and goal importance of the service class.

Goal Mode provides four types of performance goals: average response, percentile response, execution velocity, and discretionary.

- **Average response goal.** The average response is computed as the total response divided by the number of ended transactions.
- **Percentile response goal.** A percentile response performance goal means that the performance goal is "x%" of the transactions should complete within "y" time. For example, a percentile response goal could be that *90% of the transactions should complete within 200 milliseconds.*
- **Execution velocity goal.** An execution velocity goal is a measure of how fast work should run when the work is *ready* to run, without being delayed waiting for WLM-managed resources. Delays for WLM-managed resources include delays waiting for access to a CPU, delayed waiting for access to processor storage, or delayed waiting for server address space creation.

With OS/390 Release 3, execution velocity can optionally include delays waiting for non-paging DASD I/O operations. Non-paging DASD I/O delays include IOS queue delays, subchannel pending delays, and control unit queue delays. (Note that

DASD disconnect time is not included in the execution velocity delay calculations.)

With OS/390 (Version 2 Release 4), execution velocity can optionally include delays waiting for an initiator (with batch jobs assigned to WLM-managed job classes).

- **Discretionary goal.** A discretionary goal simply tells the Workload Manager to process the corresponding workload on a "discretionary" basis. The Workload Manager may allocate resources to discretionary work to optimize resource usage. However, the Workload Manager normally will not allocate resources to discretionary work if allocating the resources would cause any service class to miss its performance goal.

Users specify how important it is that the performance goals are met. Five levels of importance can be specified: *Importance 1* (very high) to *Importance 5* (not important). These importance values (1 to 5) may be specified for service classes that have a response goal or an execution velocity goal.

The Workload Manager internally defines **Importance 0** to represent system tasks (the most important work) and **Importance 6** to represent discretionary work (the least important work)¹.

A service class may be broken into multiple *performance periods*. Each service class has Period 1 automatically defined. Optionally, installations can define up to seven additional performance periods (although typically no more than two or three additional performance periods are defined). Each performance period has its own performance goal, goal importance, and duration (except that the duration may not be specified for the last performance period).

¹The goal importance for discretionary work is recorded as **Importance 0** in SMF Type 72 records because Workload Manager developers felt that there was no value in reporting Importance 6 externally. From an external view, the lowest importance (Importance 6) is implicit in the fact that the work is discretionary.

Detailed information about each service class period is recorded in SMF Type 72 records. This information includes the performance of work executing in the service class period, resources used by the address spaces in the service class period, and execution delays to work executing in the service class period.

2.0 TRANSACTION SERVICE CLASSES

With CICS/ESA Version 4.1 or IMS/ESA Version 5, the WLM workload classification scheme allows installations to classify CICS and IMS **transactions** into workload categories. The workload categories can be assigned to *transaction service classes*. Installations can specify a response goal and goal importance for the CICS or IMS transactions executing in each transaction service class.

The Workload Manager monitors the performance of the service classes to which the transactions are assigned. The WLM can manage the allocation of system resources to **address spaces** (CICS regions, IMS regions, and other address spaces supporting the CICS or IMS transactions) based on how well the transactions meet the response goals specified for their service classes.

Appreciating the distinction between transactions and the address spaces processing the transactions is important. Transactions have response goals and do not consume resources; address spaces processing the transactions do not have response goals, but consume resources while processing the transactions.

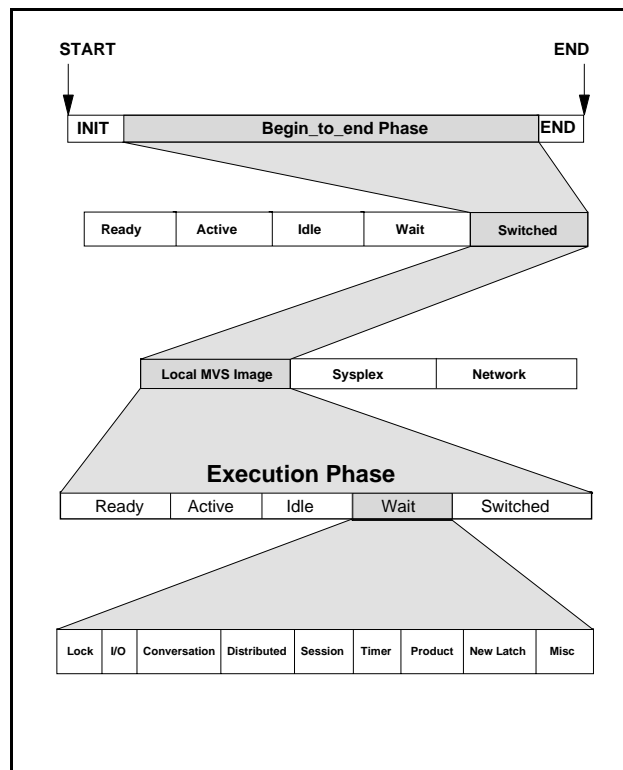
Multiple performance periods may not be defined for transaction service classes. Multiple performance periods may not be defined for service classes describing the address spaces processing the transactions.

IBM provides a standardized interface between the Workload Manager and subsystems (such as CICS and IMS). Both CICS (with Version 4.1) and IMS (with Version 5.1) take advantage of the interface to provide detailed information about the internal processing of transactions.

As shown by Figure 1, subsystems report two separate views of the transactions: the **begin_to_end phase** and the **execution phase**².

- **Begin_to_end phase.** The begin_to_end (BTE) phase starts when MVS has classified the transaction. This classification action is normally requested in a CICS Terminal Owning Region (TOR) for CICS or when the transaction is passed to the Message Processing Region for IMS.

²Note that some CICS transactions 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.



WLM VIEW OF SUBSYSTEM TRANSACTIONS
Figure 1

- **Execution phase.** The execution phase starts when CICS or IMS has started an application task to process the transaction. Processing the transaction is normally done in a CICS Application Owning Region (AOR)³ or IMS Message Processing Region.

Within both the BTE Phase and the Execution Phase, CICS provides information describing the state of the transaction: Ready state, Active state, Idle state, Wait state, and Switched state⁴. IMS provides information about the state of the transaction only in the Execution Phase: Active state, Idle state, Free State, Waiting for I/O, and Waiting for Lock.

This information is collected by RMF and written to SMF Type 72 records.

3.0 WORKLOAD ACTIVITY REPORTS

The RMF Post-Processor analyzes the SMF Type 72 records and produces the Workload Activity Reports.

³The transaction might be switched to an IMS region (the IMS region would report IMS-related states, and this information would be recorded as an additional Execution Phase).

⁴See the companion paper "Evaluating and improving CICS performance in MVS (Goal Mode)" for a discussion of the information provided by CICS.

The Workload Activity Report contains information describing each service class period (and report class) defined to the Workload Manager.

$$\frac{\text{Transaction Resident Time}}{\text{RMF Interval}} \text{ or } \frac{\text{R723CTRR}}{\text{SMF72INT}}$$

Part of the Workload Activity Report for Goal Mode is similar to the Workload Activity Report for pre-Goal Mode; this paper describes only the sections of the Workload Activity Report that are unique for Goal Mode. Interesting sections of the Workload Activity Report for Goal Mode have been “snipped” so the sections can easily be discussed.

The MPL value is not applicable with CICS or IMS transaction service classes.

Additionally, the reader should note that the RMF Workload Activity Report changes depending on the release of MVS. The “snips” of sections of the Workload Activity Report have been taken from various releases of MVS to illustrate different values available.

- **ENDED** is the number of address spaces or TSO transactions that completed in the RMF interval. The value is obtained from R723CRCP.

If CICS or IMS transactions have been classified by the Workload Manager, the value represents the count of completing CICS or IMS transactions. If the CICS or IMS transactions have not been classified, the value will be zero unless the CICS or IMS region terminates.

3.1 TRANSACTION SUMMARY DATA

The data shown in Snip 1 provides summary information about the address spaces or transactions executing in the service class period.

- **END/SEC** is the average number of address spaces or TSO transactions completing per second. The value is computed as

--TRANSACTIONS--		TRANSACTION TIME	HHH.MM.SS.TTT
AVG	0.00	ACTUAL	0.114
MPL	0.00	EXECUTION	0.078
ENDED	216	QUEUED	0
END/SEC	0.24	R/S AFFINITY	0
#SWAPS	0	INELIGIBLE	0
EXECUTD	215	CONVERSION	0
		STANDARD DEVIATION	0.235

$$\frac{\text{Transactions Ended}}{\text{RMF Interval}} \text{ or } \frac{\text{R723CRCP}}{\text{SMF72INT}}$$

If CICS or IMS transactions have been classified by the Workload Manager, the value represents the count of completing CICS or IMS transactions. If the CICS or IMS transactions have not been classified, the value will be zero unless a CICS or IMS region terminates.

Transaction Summary Data Snip 1

- **AVG** is the average number of transactions concurrently executing in the RMF interval. The value is computed as:

$$\frac{\text{Transaction Active Time}}{\text{RMF Interval}} \text{ or } \frac{\text{R723CTAT}}{\text{SMF72INT}}$$

For address spaces (batch jobs, started tasks, etc.) executing in the service class period, the value represents a count of ended address spaces. For TSO, the count represents the count of TSO transactions completing.

- **#SWAPS** is the number of swaps for all address spaces in the service class, during the RMF interval. The value is obtained from R723CSWC. The value is not applicable with CICS or IMS transaction service classes.
- **EXECUTD** is the number of subsystem (CICS or IMS) transactions that completed Execution Phase during the RMF interval. The value is obtained from R723CNCP. The value is not applicable with address spaces or TSO transactions.

The value will be applicable for CICS or IMS transactions only if the transactions have been classified by the Workload Manager.

The value is not applicable with CICS or IMS transaction service classes.

- **ACTUAL** transaction time is the average elapsed time of transactions ending during the RMF interval. The value is computed as:

$$\frac{\text{Transaction Elapsed Time}}{\text{RMF Interval}} \text{ or } \frac{\text{R723CTET}}{\text{SMF72INT}}$$

- **MPL** is the average number of address spaces resident in processor storage in the RMF interval. The value is computed as:

The ACTUAL transaction time will be zero if no transactions ended during the RMF interval.

Note that transaction times can be misleading if long-running address spaces execute in the service class. This is because the elapsed time of the long-running transactions will be combined with the elapsed time of short-running transactions ending in the RMF interval. Consequently, the *average* elapsed time will be larger than might be expected.

For example, suppose that many short TSO transactions ended in the same interval as a long TSO transaction ended. The *average* time of all transactions might be a misleading indicator of performance.

- **EXECUTION** time is the average time that transactions spent in execution. For address spaces, the value is computed as:

$$\frac{\text{Transaction Execution Time}}{\text{Transaction Ended Count}} \text{ or } \frac{R723CXET}{R723CRCP}$$

For subsystem transactions, the value is computed as:

$$\frac{\text{Execution Phase Time}}{\text{ExecutionPhase Ended Count}} \text{ or } \frac{R723CXET}{R723CNCP}$$

For subsystem transactions, the EXECUTION time represents *only* transactions entering Execution Phase.

- **QUEUED** transaction time differs depending on whether the report is before OS/390 (Version 2 Release 4) or with that release.

Before OS/390 (Version 2 Release 4), queued transaction time is *intended* to be the average time that transactions were queued during the RMF interval. The value is computed as:

$$\frac{(\text{Elapsed Time}) - (\text{Execution Time})}{\text{Transaction Ended Count}} \text{ or } \frac{R723CTET - R723CXET}{R723CRCP}$$

This calculation is valid for service classes containing address spaces. For example, the QUEUED time represents initiator delay time if the service class contained batch jobs.

However, the calculation is flawed for service classes representing subsystem transactions. Consider that some CICS transactions might not be routed, but spent all their time in the BTE Phase. For these transactions, their elapsed time would be reflected in R723CTET, but no time would appear in

R723CXET since they did not enter Execution Phase.

Suppose some transactions in the service class were routed and did enter Execution Phase. The QUEUED time would be calculated as the total elapsed time of *all transactions* minus the execution time of the *transactions entering Execution Phase*. Thus, the QUEUED time would include the elapsed time of all non-routed transactions.

With OS/390 (Version 2 Release 4), QUEUED time is the average time that a batch job was delayed for initiation, the time that a TSO user was delayed for part of LOGON processing, or the time that an OpenMVS transaction spent on an OpenMVS queue. This value is computed as:

$$\frac{\text{Total Queue Delay Time}}{\text{Transaction Ended Count}} \text{ or } \frac{R723CQDT}{R723CRCP}$$

With OS/390 (Version 2 Release 4), RMF prints zero as the QUEUED value for service classes representing subsystem transactions.

- **R/S AFFINITY** time is the average time that batch jobs (in a JES2 environment with WLM-managed batch initiators) were delayed for resource affinity or system scheduling. This value is provided beginning with OS/390 (Version 2 Release 4), and is computed as:

$$\frac{\text{Resource Affinity Delay Time}}{\text{Job Ended Count}} \text{ or } \frac{R723CADT}{R723CRCP}$$

- **INELIGIBLE** time is the average time that batch jobs (in a JES2 environment with WLM-managed batch initiators) were ineligible to be assigned an initiator because of operational or JES scheduling delays. This value is provided beginning with OS/390 (Version 2 Release 4), and is computed as:

$$\frac{\text{Batch Job Ineligible Time}}{\text{Transaction Ended Count}} \text{ or } \frac{R723CIQT}{R723CRCP}$$

- **CONVERSION** time is the average time that batch jobs (in a JES2 environment with WLM-managed batch initiators) spent in JCL conversion. This value is provided beginning with OS/390 (Version 2 Release 4), and is computed as:

$$\frac{\text{JCL Conversion Time}}{\text{Job Ended Count}} \text{ or } \frac{R723CCVT}{R723CRCP}$$

3.2 SERVICE CLASS PERFORMANCE DATA

The data shown in Snip 2 shows the overall performance information provided for service classes.

	---RESPONSE TIME---	EX	PERF	AVG	--USING%--	
	HH.MM.SS.TTT	VEL	INDX	ADRSP	CPU	I/O
GOALS		20.0%				
ACTUALS		13.4%	1.5	2.6	10.5	2.3

**Service Class Performance Data
Snip 2**

- **RESPONSE TIME.** The response time is shown for service class periods with a response goal.

For an average response goal, the response goal and “AVG” will be shown on the GOALS. The ACTUALS line shows the average response time.

For a percentile response goal, the GOALS line will have the response goal, followed by the percent of transactions that should complete within the response goal. The ACTUALS line will show the achieved response and the percent of ended transactions that met the response goal.

- **EX. VEL.** The execution velocity is provided for service classes representing address spaces. The execution velocity goal is shown on the “GOALS” line if an execution velocity goal had been specified for the service class period. The execution velocity achieved is shown on the “ACTUALS” line regardless of whether an execution velocity goal or a response goal had been specified.

“N/A” is shown on the “ACTUALS” line for subsystem transaction service classes. This annotation is shown because subsystem transactions do not consume system resources (the resources are consumed by the address spaces processing the transactions).

- **PERF INDX.** The Workload Manager periodically assesses the performance of each service class, comparing the performance **specified** for the service class against the performance goals **achieved** by the service class⁵. The comparison of performance achieved is accomplished by computing a *Performance Index* for each service class.

A Performance Index less than one implies that a performance goal **has** been met, while a Performance Index greater than one implies that a

⁵For *discretionary* goals, the Performance Index is arbitrarily established as 0.81 by the Workload Manager. The 0.81 Performance Index value means that service class periods with a discretionary goal are always considered to be meeting their goal from the view of the Workload Manager.

goal has **not** been met. A Performance Index of exactly one implies that the performance goal has been **exactly** met.

- **AVG ADRSP.** The Average Address Spaces is the number of address spaces contributing to samples shown in the address space Execution Delay information.

The Average Address Spaces can be larger than the Average Transactions or Average MPL. For example, this situation could occur if address spaces were idle for long periods.

The AVG ADRSP is not shown for subsystem transaction service classes (e.g., CICS or IMS transactions).

- **USING CPU %.** The percent CPU Using is the percent of samples taken when some dispatchable unit was using the CPU.

The USING CPU is not shown for subsystem transaction service classes (e.g., CICS or IMS transactions), since subsystem transactions do not consume system resources.

- **USING I/O %.** The percent I/O Using is the percent of samples taken when some dispatchable unit was using the non-paging DASD I/O. Within this context, the dispatchable unit was using the non-paging DASD in either a device connect state (transferring data) or a device disconnect state (seeking and latency).

The USING I/O is not shown for subsystem transaction service classes (e.g., CICS or IMS transactions).

3.3 ADDRESS SPACE EXECUTION DELAYS

The System Resources Manager (SRM) maintains control blocks for each service class period to describe the resources used by the service class period, delays to the service class period, etc. The Workload Manager periodically examines the control blocks and acquires samples⁶ describing the state of each dispatchable unit of an address space (that is, each TCB or SRB).

The Workload Manager accumulates the samples into counters that describe the state of the service class period⁷.

⁶The sampling is done every 250 milliseconds. The sampling interval is recorded in SMF Type 72 records (R723MTVL).

⁷Address spaces can have more than one dispatchable unit (that is, more than one TCB or SRB). If an address space did have more than one dispatchable unit, the Workload Manager would accumulate state samples to describe the state of each dispatchable unit.

The data shown in Snip 3 provides information about execution delays experienced by address spaces in the service class period. The execution delays are shown as a percent of the total execution and delay samples accumulated for address spaces executing in the service class period during the RMF interval .

----- EXECUTION DELAYS % -----												
TOTAL	CPU	CAPP	I/O	SWIN	MPL	-AUXILIARY	PAGE	DELAYS	%	FROM-	PRIV	COMM
						XMEM	VIO	SHSP	EHSP			
67.7	67.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Address Space Execution Delays Snip 3

- **TOTAL.** The total percent is the sum of the following execution delay percentages.
- **CPU.** The percent CPU delay is the percent of samples taken when some dispatchable unit was ready to use the CPU, but was denied access because of dispatching priorities.
- **CAPP.** The percent CPU Capping delay is the percent of samples taken when some dispatchable unit was ready to use the CPU but the WLM had implemented CPU Capping for all dispatchable units in the service class period.

CPU Capping is implemented when the maximum CPU service units have been consumed for the Resource Group to which the service class was assigned, and the Workload Manager had marked service classes associated with the Resource Group as “non-dispatchable” for some cap-slice intervals⁹.

This delay does not necessarily mean that 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

To simplify the discussion, we shall refer to the sample states as being the "state" of the address space, with readers understanding that multiple dispatchable units could simultaneously be waiting on multiple events, or could be using the processor. Thus, for any single sampling, a swapped-in address space could be counted in more than one state. Only one sample per address space would be taken for swapped-out address spaces.

⁹Not all of the execution delays shown in Snip 3 would be displayed by the RMF Post-Processor. The RMF Post-Processor displays only those execution delays accounting for a significant percent of execution delay. This feature was implemented to minimize the “clutter” of the report. Additionally, the feature allows the Workload Activity Report to reflect additional execution delays as additional delay categories are added to the SMF Type 72 records.

¹⁰With OS/390((Version 2 Release 6), the CAPP delay can include capping delay because of **discretionary goal management**. With discretionary goal management, the Workload Manager can cap overachieving work to allow discretionary work to run, even though the overachieving work had not been assigned to a Resource Group.

service class had been assigned to the Resource Group.

The CPU Capping delay state applies only to dispatchable units (TCBs and SRBs) on the Dispatcher queue awaiting dispatch. The CPU Capping delay state does not apply to dispatchable units waiting for some other event (waiting for I/O, waiting for ENQ, etc.).

- **I/O.** The percent I/O delay is the percent of samples taken when work was delayed for non-paging DASD I/O. The I/O delay includes IOS queue, subchannel pending, and control unit queue delays.
- **SWIN.** The percent Swap-In delay is the percent of samples taken when the address space was delayed on swap-in (the swap-in had started, but had not completed). These delays should nonexistent or extremely infrequent, unless serious problems exist with the auxiliary storage subsystem.
- **MPL.** The percent MPL delay is the percent of samples taken when an address space was ready to be swapped in, but that the SRM had not initiated a swap-in because of target MPL constraints associated with the domain. The SRM creates a domain for each service class period. Thus, the Workload Manager had imposed MPL constraints on the service class period to which the address space belonged.
- **PRIV.** The percent Auxiliary Page delay from Private is the percent of samples when an address space experienced page faults in the private area, and the page faults were resolved from auxiliary storage.
- **COMM.** The percent Common Area page-in from auxiliary storage is the percent of samples when the address space experienced page faults in the Common Area, and the page faults were resolved from auxiliary storage.
- **XMEM.** The percent Cross-memory page-in from auxiliary storage delay is the percent of samples when the address space experienced page faults in cross-memory access, and the page faults were resolved from auxiliary storage.
- **VIO.** The percent VIO page-in from auxiliary storage is the percent of samples when the address space experienced page faults in VIO, and the page faults were resolved from auxiliary storage.
- **SHSP.** The percent Standard hyperspace page-in from auxiliary storage delay is the percent of samples when the address space experienced page faults in standard hyperspace, and the page faults were resolved from auxiliary storage.

- **EHSP.** The percent ESO hiperspace page-in from auxiliary storage is a computed value. 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 explanation¹⁰:

"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 estimate is needed to properly manage the amount of expanded owned by the address space to the address space's goal."

The following delays are not shown in Snip 3, but would appear in the Workload Activity Report:

- **OTHR.** The percent Other delay is the percent of samples when the Workload Manager was unable to identify the cause of delay. In practice, this means that the delay was caused by something over which the SRM had no control (e.g., I/O operations not under control of the SRM, ENQ delay, etc.).
- **IDLE.** The percent Idle Delay is the percent of samples when the address space was idle (the address space was in STIMER wait, TSO terminal wait, or APPC wait; or an initiator was idle).

3.4 SUBSYSTEM RESPONSE PERCENTAGES

Subsystems communicate with the Workload Manager using Workload Manager Services macros¹¹. When a subsystem is started under Goal Mode, the region requests that MVS allocate *Performance Blocks* for transactions in the region. As transactions are processed, the subsystem will provide MVS with information about the "state" of the transaction by issuing

¹⁰IBM TALKLink RMF FORUM appended at 15:39:18 on 95/05/29 GMT (by YOCOM at KGNVMC) Subject: Workload Activity Report. Reproduced with permission of the author.

¹¹These macros are described in IBM's *Programming: Workload Management Services* document.

the IWMMCHST ("Change State of Work Request") macro. MVS simply sets bits in the Performance Blocks to indicate the state of the transaction.

The Workload Manager periodically examines the Performance Blocks and acquires samples describing the state of the transactions in the service class. The samples are accumulated into counters describing each state. RMF writes the count of samples in SMF Type 72 records.

The RMF Post-Processor converts the count of samples taken during the RMF interval into a percent of transaction elapsed time. RMF calculates the divisor necessary to produce the percent based on the elapsed time of ended transactions. As will be described later, the calculation can produce surprising results.

3.5 SUBSYSTEM RESPONSE INFORMATION

The Workload Activity Report contains response information provided by the subsystems that exploit the WLM interface. Snip 4 shows response information provided for transactions processed by the subsystem.

SUB TYPE	P	TOTAL	ACTIVE	READY	IDLE
CICS BTE		93.4	10.2	0.0	0.0
CICS EXE		67.0	13.2	7.1	0.0

**Subsystem Response Information
Snip 4**

- **SUB TYPE** is the subsystem type that produced the samples reflected by the percentages on the specific line. Currently, only CICS and IMS have exploited the WLM interface to provide information about subsystem delays to transaction processing¹². Consequently, either CICS or IMS will be shown as the subsystem type.
- **P** is the phase represented by the percentages on the line. For CICS, the phase will be BTE (begin_to_end phase) or EXE (execution phase). For IMS, only EXE will be reflected since IMS does not generate samples for a BTE phase.
- **TOTAL** is the total percent of samples represented by the line.
- **ACTIVE** is the percent of the transaction elapsed time that a program was executing on behalf of the transaction, **from the perspective of the**

¹²The interface is generalized, so other subsystems (e.g., DDF or SMS) could easily provide subsystem response and delay information.

subsystem. The fact that the subsystem reports "Active" state does not mean that the programs in the subsystem are processing the transaction.

MVS allocates CPU cycles based on MVS dispatching priority, and the subsystem (e.g., a CICS region) may be denied access to a CPU. For example, an address space with a higher dispatching priority could have preempted CICS. Consequently, the subsystem could be waiting for access to a CPU and not actually executing, although the subsystem would have reported that the transaction was in Active state.

- **READY** is the percent of the transaction elapsed time that a program was ready to execute on behalf of a transaction, but that the subsystem has given priority to another transaction.
- **IDLE** is the percent of the transaction elapsed time that no work requests were ready to run in the service class.

3.6 SUBSYSTEM RESPONSE DELAYS

The subsystem can report a variety of response delays¹³, but not all subsystems provide information for each delay category. The developers of each subsystem exploiting the WLM interface decide which delays to report and decide which modules in the subsystem report the delays.

Snip 5 shows the calculated percent of transaction response delay waiting for various categories.

-----WAITING FOR-----										
LOCK	I/O	CONV	DIST	LOCAL	SYSPL	REMOT	TIMER	PROD	MISC	
0.0	0.0	83.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	46.7	0.0	

**Subsystem Response Delays
Snip 5**

- **WAITING FOR LOCK** is the percent of the transaction elapsed time that a program was waiting for a lock. Both CICS and IMS inform the WLM about this wait condition.
- **WAITING FOR I/O** is the percent of the transaction elapsed time that a program was waiting for some I/O request. The program could be waiting on an actual I/O operation or could be waiting on some

function related to the I/O request. Both CICS and IMS inform the WLM about this wait condition.

- **WAITING FOR CONV** is the percent of the transaction elapsed time that a program was waiting for a response in a conversation mode. Only CICS informs the WLM about this wait condition.

For CICS, the wait condition means that CICS has shipped the transaction from a TOR to an AOR, or to IMS. The "SWITCHED TO" information on the Workload Activity Report will show whether the transaction was switched in the local system, somewhere in the sysplex, or somewhere in the network.

- **WAITING FOR DIST** is the percent of the transaction elapsed time that a program was waiting some function or data must be routed before resumption of the work request. Neither CICS nor IMS use this wait condition.

- **WAITING FOR LOCAL** is the percent of the transaction elapsed time that a program was waiting for a session to be established on the current MVS image. Only CICS informs the WLM about this wait condition.

- **WAITING FOR SYSPL** is the percent of the transaction elapsed time that a program was waiting for a session to be established somewhere in the sysplex. Only CICS informs the WLM about this wait condition.

- **WAITING FOR REMOTE** is the percent of the transaction elapsed time that a program was waiting for a session to be established somewhere in the network. Only CICS informs the WLM about this wait condition.

- **WAITING FOR TIMER** is the percent of the transaction elapsed time that a program was waiting for expiration of a timer. Only CICS informs the WLM about this wait condition.

- **WAITING FOR PROD** is the percent of the transaction elapsed time that a program was waiting for another product to provide some service. Only CICS informs the WLM about this wait condition.

- **WAITING FOR MISC** is the percent of the transaction elapsed time that a program was waiting, but the work manager could not identify the cause of the wait. Only CICS informs the WLM about this wait condition.

CICS documentation calls this wait condition Miscellaneous Waits (or MISC). Table 18 of the *CICS Problem Determination Guide* provides detailed information about each module in CICS that

¹³With OS/390 (Version 2 Release 4), the WLM Programming Services Manual lists 15 resources that subsystems could be waiting for. Existing exploiters of the WLM interface do not use all wait categories, and RMF does not yet report all categories.

issues a call to the WLM with a MISC reason, and shows the resource that CICS is waiting on.

With OS/390 (Version 1, Release 3), the Wait for New Latch is recorded in SMF Type 72 records, but this wait condition is not reported by RMF.

Additionally, with OS/390 (Version 2, Release 4), the IWMMCHST macro permits several new resource values for the Waiting State, to indicate waiting on BUFFER_POOL_IO, waiting on BUFFER_POOL_CF, waiting on BUFFER_POOL_CF_IO, and waiting on CF_IO. The Workload Manager Services Macro document describes these parameters. At present, RMF neither records these values in SMF Type 72 records nor reports the values in the Workload Activity Report.

4.0 SUBSYSTEM RESPONSE PERCENTAGES

As mentioned earlier, RMF calculates the divisor for the subsystem response percentages. This section describes the calculation and shows why the calculation can produce surprising results.

The elapsed time of ended transactions is contained in the SMF variable R723CTET. RMF multiplies the value in R723CTET by the number of samples taken per second (currently, four samples are taken each second). The resulting value is the total number of samples that would be taken during the elapsed time of transactions ending in the RMF interval.

Each sample count (Active count, Ready count, Waiting count, etc.) is divided by the calculated total number of samples to yield a percent of transaction time in each sampled state.

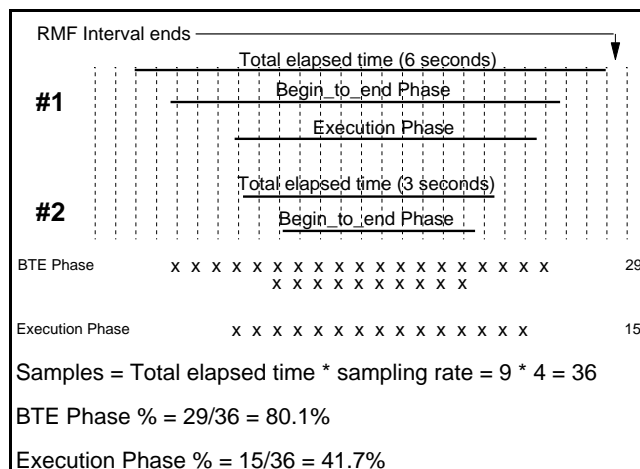
4.1 RESPONSE PERCENT CALCULATION

Figure 2 illustrates the subsystem response percentage calculation. Shown in Figure 2 is the sampling process for the BTE Phase and Execution Phase, and the subsystem response percentage calculation of the BTE Phase and Execution Phase.

Figure 2 shows two example subsystem transactions executing in a service class. Transaction #1 has a total elapsed time of six seconds, and has both a BTE Phase and an Execution Phase. Transaction #2 has a total elapsed time of three seconds, and has only a BTE Phase. A total of nine seconds of elapsed time was recorded for the two transactions ending in the RMF Interval.

The dotted vertical lines represent WLM sampling intervals, with samples taken four times per second. Notice that the horizontal lines representing the start and stop time of the transactions and their BTE or Execution Phases do not start and end exactly at the sample

interval. This is deliberate to illustrate that there is some error implicit in the sampling process.



Subsystem Response Percentage Calculation

Figure 2

The samples for the two example transactions are accumulated into counters for the service class. Shown on the right of Figure 2 is a summary of the samples as the RMF Interval ends and the SMF Type 72 record is written. Twenty-nine samples were taken while Transaction #1 and Transaction #2 were in the BTE Phase. Fifteen samples were taken while Transaction #1 was in the Execution Phase (Transaction #2 did not enter the Execution Phase).

Figure 2 illustrates that RMF would calculate that 36 samples should have been taken for the two transactions. This value would be used by RMF as the divisor for computing the percent of transaction response time spent in the BTE Phase and the Execution Phase for transactions executing in the service class. As shown, 80.1% of the transaction time was spent in the BTE Phase and 41.7% of the transaction time was spent in the Execution Phase.

4.2 RESPONSE PERCENT MISCALCULATION

The samples of transaction state are taken dynamically and the count of samples is recorded at the end of the RMF interval. These samples are divided by a computed value intended to represent the elapsed time of transactions from which the samples were taken. However, the calculation of the divisor is based on the elapsed time of ended transactions within the RMF interval for which the calculations are made.

The RMF Post-Processor would compute 8.3% as the response percentage in BTE Phase for this service class, and would compute 4.2% as the response percentage in the Execution Phase for this service class.

Only the BTE Phase and Execution Phase percentages are shown in Figure 3-5. Other percentages are calculated similarly, and could be miscalculated.

IBM documents contain many examples in which the percent calculations result in extremely large (significantly more than 100%) values. For example, the *CICS Performance Guide* shows an example of 78,000% of response time in a specific category. Users often observe similar values when examining RMF Workload Activity Reports.

5.0 CONCLUSION

From the discussion about response percent miscalculation, it would be easy to conclude that the response delay percentages have little value. Quite clearly, no transaction response delay can account for more than 100% of the transaction elapsed time. However, this conclusion would be incorrect!

Even though the response delay percentages might be incorrect, the **relative values of the percentages** give insight into the delays experienced by the transactions. The WLM samples are used as the numerator, and these samples are valid indicators of the delays experienced by the transactions.

It is easy to identify the most significant response delays, even though the actual percentages are meaningless. Performance analysts can focus on the delay categories represented by large values.

With the data available in MVS (Goal Mode), IBM has provided a wealth of information about the performance of workloads executing under MVS, and about the constraints to improved performance. The information now available as a standard part of RMF reporting previously was available only with sophisticated (and relatively expensive) commercial tools.

6.0 REFERENCES

OS/390 V2R5 Planning: Workload Management

OS/390 V2R5 Workload Management Services

OS/390 V2R5 MVS System Management Facilities

OS/390 V2R4 RMF Report Analysis

CICS/ESA Version 4.1 Performance Guide

CICS/TS Release 1.2 Performance Guide

Steve Grabarits and Gail Whistance, "Pop the Hood on Workload Manager", SHARE Technical Conference, August 1998, Session 2513.