



John Burgess  
Manuela Mandelli

# IBM CICS Performance Series: Effective Monitoring for CICS Performance Benchmarks

## Introduction

The aim of this paper is to show tools and techniques that can be used to understand the performance characteristics of an IBM® CICS® performance benchmark. It describes why it is important to understand whether other work on your systems can interfere with the system under test. It looks at an example CICS Transaction Server (CICS TS) configuration and describes how monitoring tools, including IBM Resource Measurement Facility™ (IBM RMF™) and Workload Manager (WLM) can be used to collect and report performance data.

## Achieving consistent, repeatable CPU usage in benchmarks

Before we describe how to collect performance data, it is important to explain how, unless you totally dedicate hardware for a benchmark, the CPU that is used can vary each time that the benchmark is run. It might be hard to get repeatable results. This statement is true for benchmark comparisons and also for CPU usage comparisons after a CICS upgrade.

A common misconception is that the same transaction running the same application programs using the same amount of business logic uses the same amount of CPU time each time that it is run. It probably executes the same number of physical instructions; however, the number of required cycles to execute those instructions can vary, depending on the state of the on-chip caches. Two identical instructions can use varying numbers of cycles to complete, depending on the level of cache or real storage that the data or instructions were retrieved from.

Code in a CICS region running, for example, on a zEC12, is affected by the state of cache hierarchy if the following conditions are true:

- ▶ It is running on a single logical partition (LPAR) of z hardware with any other type of work executing in other address spaces on the same LPAR.
- ▶ Work is running on any other LPAR sharing the same central processors (CPs) as the LPAR where the CICS region resides.
- ▶ Work is running on any other LPAR that is using CPs that are on the same processor unit (PU) chip as the CPs that the LPAR running CICS is using because CPs on the same PU Chip share a level 3 cache. CPs on the same PU Chip share a level 3 cache even if the LPAR under test has dedicated CPs.
- ▶ Work is running on any other LPAR that is using CPs that are on the same book as the CPs that the LPAR running CICS uses because CPs on the same book share a level 4 cache and the same real storage. CPs on the same book share a level 4 cache and the same real storage even if the LPAR under test has dedicated CPs.

Figure 1 shows the memory hierarchy of a zEC12.

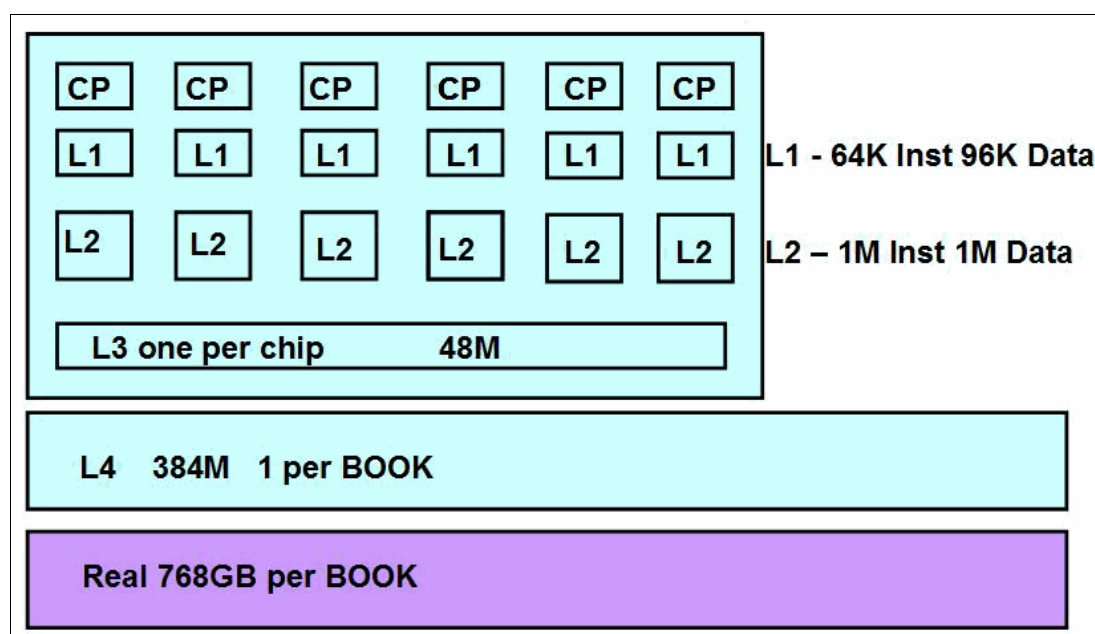


Figure 1 zEC12 cache layout

Figure 1 shows up to six CPs on a PU chip, six PU chips in a book, and four books in a central processor complex (CEC). Not all CPs are configurable as general-purpose CPs. CPs are associated with an L1 cache and an L2 cache and shared use of the L3 and L4 caches. The deeper into the cache hierarchy the instruction needs to go for its data, the more CPU cycles it uses. If the cache that programs are using is getting disrupted by other programs even from other LPARs, the data must be retrieved from higher-level caches or real storage, therefore varying the cost of these instructions.

This depth into the memory hierarchy, to which a workload that is running on an LPAR needs to go to retrieve its data, is referred to as *Relative Nest Intensity* (RNI). RNI is a metric that is now used to select the most relevant workload when you use the IBM Large Systems Performance Report (LSPR) to calculate the effects of hardware migration. Historically, LSPR workload capacity values used application names, such as CICS and IBM IMS™.

However, because capacity performance was always more closely associated with how a workload uses and interacts with a particular processor hardware design and more importantly its caches, this new RNI metric was introduced with the ability for clients to calculate this value for their own workloads.

For more information, see this website, which provides a more detailed description of LSPR and RNI:

<https://www-304.ibm.com/servers/resourceLink/lib03060.nsf/pages/lsprwork?OpenDocument>

On IBM z10™ and later machines, the ability to collect instrumentation data that describes the interaction with memory was introduced. This Hardware Instrumentation (HIS) can be used to produce IBM z/OS® file system (zFS) data and System Management Facilities (SMF) 113 records. This data can be used to provide a detailed insight into the workload characteristics. Also, this data can be used to determine in a benchmark that the interaction between the software and hardware was the same in two comparative runs and that the results were not affected by outside influences (Figure 2).

For more information about how to gather and use this HIS data, see the IBM Redpaper™, *Setting Up and Using the IBM System z® CPU Measurement Facility with z/OS*, REDP-4727-00.

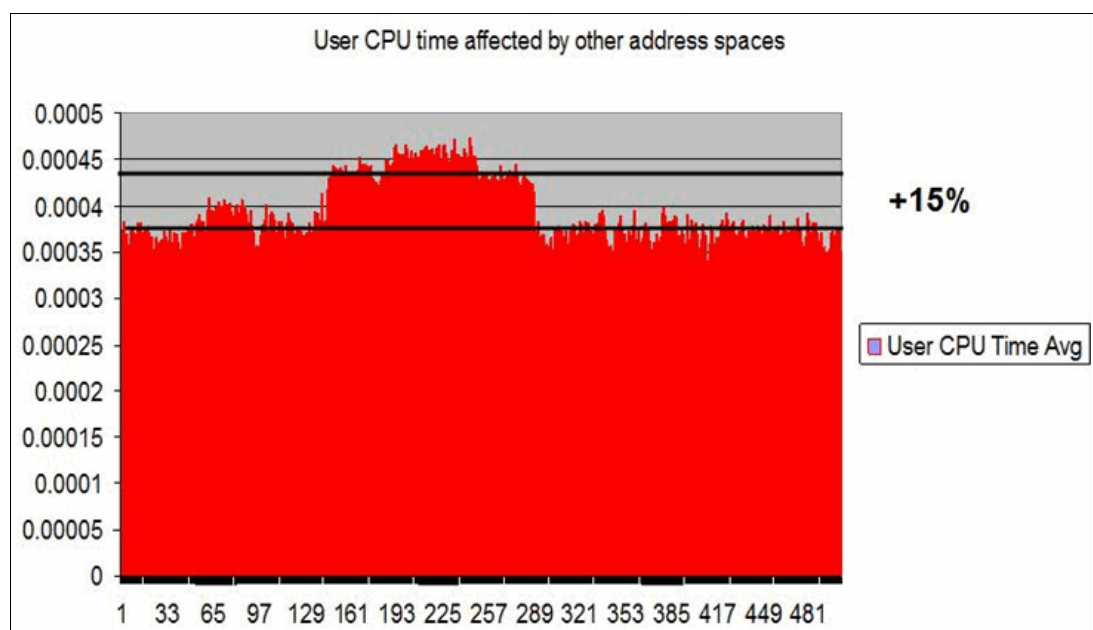


Figure 2 Other work that affects CICS CPU per transaction

Figure 2 shows the effect on CICS CPU/transactions when other batch-type work in separate address spaces is started on the same LPAR. The other work, which was introduced for a short time, causes more contention in the cache hierarchy and the cycles per instruction (CPI) increases for the CICS workload. In this example, there was an increase of up to 15% more CPU that was used during that period.

In summary, to get consistent and repeatable CPU usage from a benchmark comparison, it is important to have a configuration that cannot be affected by other work anywhere on the system, either on the LPAR under test or any other LPAR on the machine.

In reality, not everyone can dedicate hardware to a benchmark. So, in this case, take sufficient measurements to establish a trend, especially in comparisons before and after a CICS upgrade where a trend from before and after must be established so that extremes or outlying cases can be eliminated.

## CICS Transaction Server (CICS TS) configuration

For demonstration, we chose the following CICS configuration (Figure 3) to use as a performance benchmark.

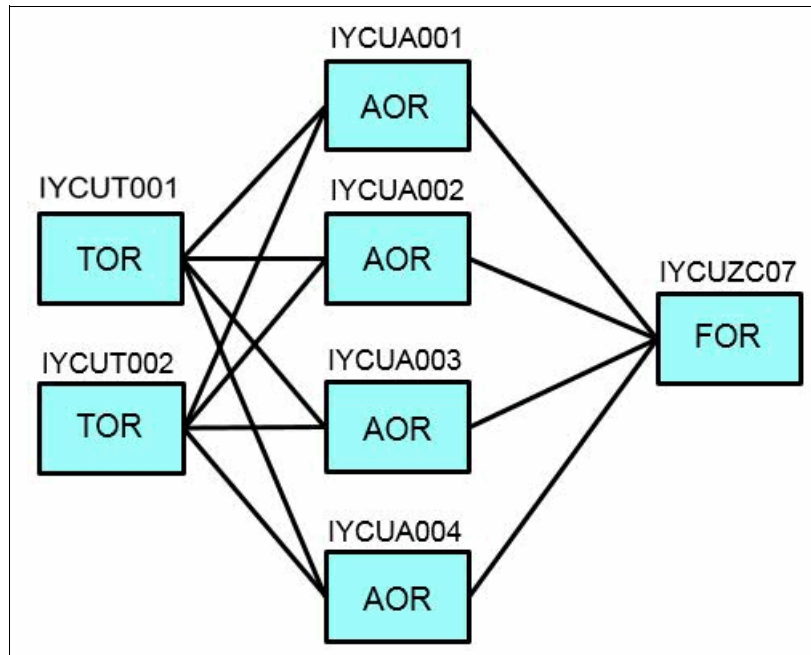


Figure 3 CICS configuration

As shown in Figure 3, there were two terminal-owning regions (TORs) that both route all transactions to all of the four application-owning regions (AORs). All AORs then function-ship all VSAM file requests to a single data-owning or file-owning region (FOR). The application identifiers (APPLIDs) for the regions are also shown in Figure 3.

For reference with the WLM setup later, the job names of the TORs are CICST001 and CICST002; the job names of the AORs are CICS001, CICS002, CICS003, and CICS004; and the job name for the FOR is CICS2A07.

## CICS TS application

The application that is used for this benchmark has a relatively short path length in terms of business logic.

It has the following characteristics:

- ▶ COBOL/VSAM.
- ▶ All transactions arriving in the TORs are routed to the AORs.
- ▶ Routing is performed by using IBM CICSplex® System Manager (SM).

- ▶ A few transactions start in the AOR by way of a Start from a routed transaction.
- ▶ On average, 12 file requests occur per transaction.
- ▶ All file requests are function-shipped to the FOR, 69% Read, 10% Read for Update, 9% Update, 11% Add, and 1% Delete. Long-running mirrors are used.
- ▶ To avoid any transaction affinities, all temporary storage requests use a temporary storage data sharing server.

## Hardware and software configuration

This section describes the hardware and software that are used during this benchmark.

The following hardware is used:

- ▶ zEC12 2827-799 model HA1
- ▶ LPAR with three dedicated CPs for the system under test
- ▶ LPAR with three dedicated CPs for the network driver
- ▶ Internal Coupling Facility with ICP links

The following software is used:

- ▶ CICS Transaction Server V5R2
- ▶ z/OS V2R1
- ▶ IBM Workload Simulator for z/OS (WSim)

## Summary of performance monitoring tools

During the benchmark sampling period, the following tools were used.

### RMF Monitor I

RMF Monitor I was used to record system resource usage, including CPU, DASD, and storage. Also, RFM Monitor I was used with the WLM configuration to record the CPU, transaction rates, and response times for CICS service classes and report classes.

SMF records 70 - 79 are written on an interval basis. They can be post-processed by using the RMF utility program ERBRMFPP.

### RMF Monitor III

RMF Monitor III was used to record the coupling facility activity for the logger and temporary storage structures. SMF records 70 - 79 are written on an interval basis. Also, they can be post-processed by using the RMF utility program ERBRMFPP. RMF Monitor III can be used on an interactive basis and the data can be written to VSAM datasets for later review.

### CICS TS statistics

CICS statistics were used to monitor and report CICS resource usage, including CPU, storage, file accesses, the number of requests that were transaction-routed, and the number of requests that were function-shipped.

With CICS interval statistics, most of the counters are reset at the start of the interval so that any resource consumption that is reported relates only to the observed measurement period. Interval statistics can be activated by using **CEMT SET STATISTICS**. However, when you set this interval, the first interval can be adjusted to a shorter time so that all the intervals are synchronized to the **STATEOD** parameter. For example, if you use **CEMT** to set the interval to 15 minutes at 10 past the hour, the first interval expires in 5 minutes so that all future intervals line up on 15-minute wall clock boundaries. The values in this first report can also be associated with a much longer period, depending on the time of the last reset.

Another alternative to using interval statistics is to use **CEMT** to reset the counters and then at the end of the measurement period, use **CEMT** to record all the statistics. Resetting the statistics requires a change of state from ON to OFF or from OFF to ON. To ensure that this change happens, the commands that are shown in Figure 4 show an example of resetting the statistics in one of the AORs in this configuration.

```
F CICS001,CEMT SET STAT OFF RESET  
F CICS001,CEMT SET STAT ON RESET  
    Measurement period is between the RESET and the RECORD  
F CICS001,CEMT PERFORM STAT ALL RECORD
```

*Figure 4 Collecting CICS statistics*

Regardless of whether the statistics are ON or OFF, when a **PERFORM RECORD** is issued, a statistics record is written.

CICS statistics are written as SMF 110 subtype 2 records. They can be post-processed by using the CICS statistics utility program, DFHSTUP, or CICS Performance Analyzer (CICS PA).

## CICS TS Performance Class Monitoring

When CICS Performance Class Monitoring is turned on by either using **MNPER=ON** in the CICS startup parameters or by using **CEMT** or **CEMN** transactions to turn it on dynamically, a Performance Class Monitoring record is generated for every executed transaction when the transaction terminates.

The command in Figure 5 is an example of turning on CICS Performance Class Monitoring and Resource Class Monitoring for one of the AORs in this configuration.

```
F CICS001,CEMT SET MON ON PER RESRCE  
Monitoring can then be turned off using  
F CICS001,CEMT SET MON ON NOPER NORESRCE
```

*Figure 5 Collecting CICS Performance Class Monitoring data*

Each transaction's performance class record contains information about the resources that were used by that transaction, how much CPU was used on all the various task control blocks (TCBs), and information about how long it waited for different resources. Resource Class Monitoring records contain information about the individual files, temporary storage queues, and distributed program links (DPLs) that were used by transactions.

Monitoring records are written as SMF 110 Type 1 records. They can be analyzed by using CICS Performance Analyzer for z/OS.

## Hardware Instrumentation counters

Hardware Instrumentation (HIS) can be used to collect Instruction samples and counters. Only counters were used for this benchmark.

HIS counters are written as SMF 113 records and to zFS. These counters contain information about key hardware events, such as the number of instructions, the number of cycles that were used, and the number of instruction/data cache misses. For more information, see *Setting Up and Using the IBM System z CPU Measurement Facility with z/OS*, REDP-4727, which provides a good description of how to set up and use HIS:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4727.pdf>

## z/OS Workload Manager (WLM) and CICS

z/OS Workload Manager provides a solution for managing workload distribution, workload balancing, and distributing resources to competing workloads.

### WLM service definition constructs

Figure 6 shows the main components of a z/OS WLM service definition.

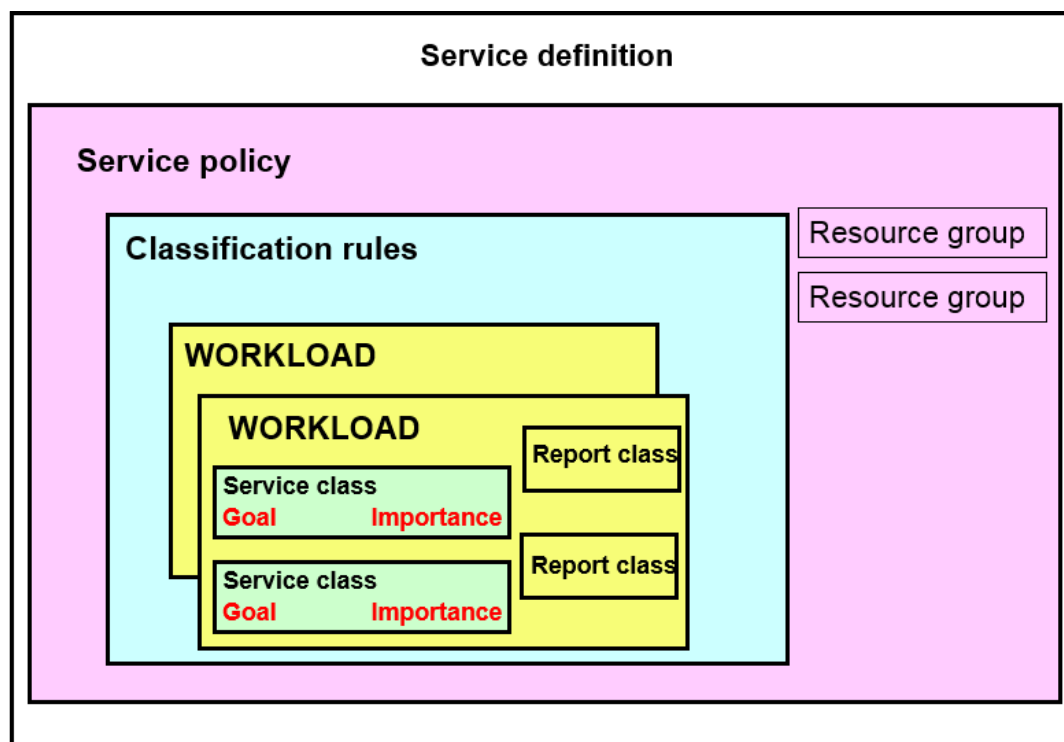


Figure 6 WLM service definition constructs

The main components are described:

- ▶ Service definition

The service definition is composed of a set of service policies. There is only one service definition for a sysplex.

- ▶ Service policies

A service policy applies to all of the work running in a sysplex. A service policy is a collection of performance goals that are described in a set of service classes.

Multiple service policies with different objectives can exist, but only one can be active at a time.

- ▶ Resource groups

Resource groups define processor capacity boundaries across the sysplex. You can assign a minimum and maximum amount of CPU service units per second to work by assigning a service class to a resource group.

- ▶ Classification rules

The classification rules link the external properties of transactions to a service class. The rules are used to assign incoming work to a service class.

- ▶ Workload

Workload is an arbitrary name that is used to group service classes for reporting and accounting.

- ▶ Service class

The service class defines the performance goals for specific types of work. The goal can be based on the response time or velocity.

- ▶ Report class

The report classes group work for reporting. They are commonly used to provide more granular reporting for subsets of work within a single service class.

WLM is used to set the service levels that you want your CICS regions and transactions to achieve. It is also a useful, relatively inexpensive method for reporting CICS CPU usage, transaction rates, and response times by way of RMF.

When work enters the system, it is classified into service classes and reporting classes based on a set of rules associated with its subsystem type.

For this benchmark, all the CICS regions were submitted as batch jobs so in this case they are all initially classified into a service class under the WLM JES classification rules. When transactions start executing in these regions, whether they stay managed to the service class that is specified in the JES classification depends on what is specified in the **Manage Region Using Goals of** parameter for that classification.

If REGION is specified, the CICS region is managed to the Velocity goal of the JES service class.

Velocity in this case is the measurement of the speed at which an address space gets dispatched on a CPU when it is in a ready state:

$$\text{Velocity} = \text{Using samples} / (\text{Using samples} + \text{Delay samples})$$

WLM samples the states of dispatchable units every 250 milliseconds, and this formula refers to these states. Using is when the dispatchable unit is using CPU or DASD I/O. Delayed is when delayed by CPU, storage, or I/O.



If TRANSACTION is specified, the regions are managed to the response time goals of the transactions that are running in them as specified in the CICS subsystem classification rules.

For this benchmark, the CICS response time goals are based on a percentile goal, for example, 80% of transactions must finish within 0.030 of a second. It is a preferred practice to set goals that are achievable. The shortest goal that can be set is 0.015.

If BOTH is specified, the regions are managed to the goals of the transactions that are running in them. However, at the same time, WLM can help achieve the velocity goal set by the JES or STC service class that is associated with these regions.

For this demonstration, we ran all our CICS transactions in the same service class and used WLM report classes and workloads to report on the different region types.

A report class can be associated with a job that was classified by the JES classification rules, in which case, CPU, I/O, and real storage are reported in that class. Also, a report class can be associated with a CICS subsystem classification based, in our case, on its CICS APPLID. This reports transaction rates and response times. If you use the same report class for both the JES and the CICS classifications, you get all of this information in a single report.

Another concept, which is called a *workload*, is an arbitrary grouping of service classes that is used for reporting and accounting. In this benchmark, we used report classes to look at TORs, AORs, and the FOR. We used the workload to accumulate all the regions into one report.

**Note:** An important factor with WLM reporting is that CICS transactions are only reported in the region in which they were originally initiated, classified, and then ended.

For instance, if in the TOR/AOR/FOR environment, 5000 transactions arrived in the TOR, these transactions are classified accordingly. If they are all routed to the AOR, they carry that classification with them. The same rule applies for any transaction that is running on behalf of these in the FOR, such as CSMI. When you look at an RMF report, the only transaction count that appears is the 5000 in the TOR. This count contains the number of transactions ended, which is the total number of transactions that were initiated in the region or regions that are represented by this report class or service class. WLM has a count of the initial transactions only, not any transaction that is started as a result of a transaction route or function-ship. In this example, where all the transactions were routed to an AOR, there is an ENDED count of 5000 and an EXCTD of 5000 in the RMF report. The EXCTD is a count of transactions that were routed to another CICS region.

The Actual and Execution response times are also shown. *Actual* refers to the overall response time of transactions and *Execution* refers to the time that is spent executing in the AORs and FOR. Figure 22 on page 19, Report Class for TORs, shows an example.

## Setting up WLM for this CICS TS configuration

Figure 3 on page 4 shows a CICS configuration that consists of two TORs, four AORs, and one FOR. By using the WLM Time Sharing Option (TSO) application, we set up WLM so that our RMF Monitor I reports show the following information:

- ▶ The transaction rate, response times, and CPU usage in the TORs
- ▶ The transaction rate, response times, and CPU usage in the AORs (this transaction count does not include those transactions that were routed from TORs)

- The transaction rate, response times, and CPU usage in the FORs (this transaction count does not include those CSMIs that are run as a result of a function-ship request from an AOR)
- The total transaction rate across all regions and total CPU used by CICS regions (Workload report)

In addition to the preceding information, WLM was also set up so that the CICS regions are transaction-managed with a percentile average response time goal of 80% of all transactions ending within 0.030 seconds.

The following windows show several of the panels that we used to achieve the required WLM settings. Figure 7 shows the TSO WLM application.

Definition name . . . . .	PLEX3	(Required)
Description . . . . .	Combined MQ & CICS definition	
Select one of the following options. . . . .		
	6	1. Policies
		2. Workloads
		3. Resource Groups
		4. Service Classes
		5. Classification Groups
		6. Classification Rules
		7. Report Classes
		8. Service Coefficients/Options
		9. Application Environments
		10. Scheduling Environments

Figure 7 TSO WLM application

Figure 7 shows the main panel for the TSO WLM application. The CICS regions that were used for this benchmark were submitted as batch jobs so these are classified by modifying the JES subsystem type under option 6, Classification Rules.

## JES subsystem classification

Figure 8 on page 11 shows the Subsystem type selection list, which is displayed after you select option 6, Classification Rules, on the previous panel (Figure 7).

Subsystem Type Selection List for Rules				Row 1 to 14	
Command ==>					
Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar					
				----- Class -----	
Action	Type	Description	Service	Report	
—	ASCH	Use Modify to enter YOUR rules	_____	_____	
—	CB	Use Modify to enter YOUR rules	WASENCLV	WASENCLV	
—	CICS	CICS Transaction classification	CICSDEF	CICS	
—	DB2	Use Modify to enter YOUR rules	_____	DB2	
—	DDF	Use Modify to enter YOUR rules	_____	_____	
—	IMS	Use Modify to enter YOUR rules	_____	_____	
—	IWEB	Use Modify to enter YOUR rules	_____	_____	
3	JES	_____	BATCH	BATCH	
—	LSFM	Use Modify to enter YOUR rules	_____	_____	
—	MQ	Use Modify to enter YOUR rules	BROKER	MQ	

Figure 8 Selecting the JES classification rules

Figure 8 shows the JES subsystem type that is selected for update.

Figure 9 shows the JES classification rules.

Action codes: A=After C=Copy M=Move I=Insert rule						
B=Before D=Delete row R=Repeat IS=Insert Sub-rule						
More ==>						
-----Qualifier-----						
Action	Type	Name	Start	Service	Report	
				DEFAULTS:	BATCH	
—	1 TN	CICST0*	—	JESTORS	CICSTORS	
—	1 TN	CICSA0*	—	CICSBTCH	CICSAORS	
—	1 TN	CICSMT*	—	JESMTOR	CICSMMT	
—	1 TN	CICSMA*	—	JESMM	CICSMAA	
—	1 TN	CICSMF*	—	JESMM	CICSMMF	
—	1 TN	CICS2A07	—	CICSBTCH	CICSFORS	

Figure 9 JES classification rules

Figure 9 shows the JES classification rules for the CICS regions that are described in Figure 3 on page 4. The two TORs start in service class JESTORS. They are reported in report class CICSTORS. The AORs start in service class CICSBTCH. They are reported in report class CICSAORS. The FOR starts in service class CICSBTCH. It is reported in report class CICSFORS.

The TORs were given a unique service class because the classification rule for these regions differs from the classification rule for the other regions. If we scroll to the right in the previous window, as shown in Figure 10 on page 12, you can see that these regions are managed by both REGION and TRANSACTION (BOTH). The AORs and FOR are managed by TRANSACTION only.

Action codes:		A=After	C=Copy	M=Move	I=Insert rule
		B=Before	D=Delete row	R=Repeat	IS=Insert Sub-rule
					<=== More
Action	-----Qualifier-----			Storage	Manage Region
	Type	Name	Start	Critical	Using Goals Of
_____ 1	TN	CICST0*	_____	NO	BOTH
_____ 1	TN	CICSA0*	_____	NO	TRANSACTION
_____ 1	TN	CICSMT*	_____	NO	BOTH
_____ 1	TN	CIC SMA*	_____	NO	TRANSACTION
_____ 1	TN	CICSMF*	_____	NO	TRANSACTION
_____ 1	TN	CICS2A07	_____	NO	TRANSACTION

Figure 10 TORs have BOTH specified

During initialization, all regions are managed to the JES subsystem classification service classes. When these CICS regions start to execute transactions, transactions are then classified according to the CICS subsystem classification rules and the regions are managed to achieve those response time goals in the CICS service classes. The TORs are managed as part of that service class (to achieve the CICS response time goals), but at the same time, their velocity goal is considered by WLM because they have BOTH specified. The AORs and the FOR are not managed to the JES service class after transactions start to execute.

## CICS subsystem classification

The CICS regions for this benchmark are managed by TRANSACTION, and in addition the TORs are managed to BOTH. The following windows show how we set up the WLM definitions for the CICS subsystem classification rules. Figure 11 shows the selection of the CICS subsystem rules.

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar				
			----- Class -----	
Action	Type	Description	Service	Report
—	ASCH	Use Modify to enter YOUR rules	_____	_____
—	CB	Use Modify to enter YOUR rules	WASENCLV	WASENCLV
<u>3</u>	CICS	CICS Transaction classification	CICSDEF	CICS
—	DB2	Use Modify to enter YOUR rules	_____	DB2
—	DDF	Use Modify to enter YOUR rules	_____	_____
—	IMS	Use Modify to enter YOUR rules	_____	_____
—	IWEB	Use Modify to enter YOUR rules	_____	_____
—	JES	_____	BATCH	BATCH

Figure 11 Selecting the CICS subsystem rules

Figure 11 shows the CICS subsystem rules that are selected for the update.

Figure 12 on page 13 shows the CICS classification rules.

Subsystem Type . . : CICS		Fold qualifier names? <u>Y</u> (Y or N)	
Description . . .		<u>CICS Transaction classification</u>	
Action codes:    A=After            C=Copy            M=Move            I=Insert rule			
B=Before        D=Delete row    R=Repeat        IS=Insert Sub-rule			
More ==>			
-----Qualifier-----		-----Class-----	
Action	Type	Name	Start
		Service	Report
		DEFAULTS: CICSDEF	CICS
_____ 1	SI	IYCUT0*	_____
_____ 1	SI	IYCUA0*	_____
_____ 1	SI	IYCUZC07	_____
_____ 1	SI	IYCUZC60	_____
_____ 1	SI	IYCUZC61	_____
		CICSJB	CICSTORS
		CICSJB	CICSAORS
		CICSJB	CICSFOR
		CICSTRAN	CICSMMT
		CICSTRAN	CICSMMT

Figure 12 CICS classification rules

Figure 12 shows that all the regions in our configuration use the same service class. All transactions are classified into service class CICSJB but the regions are grouped by type into unique report classes. Classifications are based on the APPLID of the CICS region. For example, the TORs have APPLIDs IYCUT001 and IYCUT002; so based on IYCUT0\*, all transactions that start in the TORs run in service class CICSJB and are reported in report class CICSTORS. IYCUZC07 is the name for FOR.

**Note:** In Figure 9 on page 11, the TORs, AORs, and FOR classifications for the JES subsystems show the regions reporting into the same report class names as the CICS subsystem rules in Figure 12. JES subsystems report CPU time, I/O, and real storage. CICS subsystems report transaction rates and response time. When both subsystem types point to the same report class, all data appears in one place. An example is shown later in an RMF report.

## Service classes

In this configuration, two JES subsystem service classes JESTORS and CICSBTCH exist. One CICS subsystem service class, CICSJB, exists. The AORs and FOR regions are managed only by service class CICSBTCH during the CICS initiation and until transactions start running. Then, the AORs and FOR regions switch to being TRANSACTION-managed by service class CICSJB.

Figure 13 on page 14 show the selection of service class definitions.

```

Definition data set . . . : none

Definition name . . . . . PLEX3 (Required)
Description . . . . . Combined MQ & CICS definition

Select one of the
following options. . . . . 4
1. Policies
2. Workloads
3. Resource Groups
4. Service Classes
5. Classification Groups
6. Classification Rules
7. Report Classes
8. Service Coefficients/Options
9. Application Environments
10. Scheduling Environments
11. Guest Platform Management Provider

```

Figure 13 Selecting service classes

Figure 14 lists the service classes.

Action Codes: 1=Create, 2=Copy, 3=Modify, 4=Browse, 5=Print, 6=Delete, /=Menu Bar

Action	Class	Description	Workload
—	BATCH	Unspecified batch jobs	GENERAL
—	BATCHH	service class - important batch	GENERAL
—	BROKER	test for classifying work	GENERAL
—	CICSBT	CICS JES service class	CICSCPU
—	CICSBTCH	batch service class cics	CICSCPU
—	CICSDEF	CICS default service class	CICSCPU
—	CICSJB	CICS service class for JB	CICSCPU
—	CICSTRAN	CICS service class	CICSCPU
—	CICSTST	Capped CPU usage	GENERAL
—	JESDEF	default jes service class	JESJUNK
—	JESMM	service class for Manuela	CICSCPU
—	JESMTOR	service class for Manuela TORs	CICSCPU
—	JESTORS	CICS JES service class	CICSCPU

Figure 14 Listing the service classes

Figure 14 shows a list of defined service classes.

Figure 15 on page 15 and Figure 16 on page 15 show the definitions for the CICS subsystems service class CICSJB and the JES subsystem service class JESTORS.

```

Service Class Name . . . . . : CICSJB
Description . . . . . : CICS service class for JB
Workload Name . . . . . : CICSCPU (name or ?)
Base Resource Group . . . . . : _____ (name or ?)
Cpu Critical . . . . . : NO (YES or NO)
I/O Priority Group . . . . . : NORMAL (NORMAL or HIGH)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

      -- Period -- ----- Goal -----
Action # Duration Imp. Description
-----
  1      2      80% complete within 00:00:00.030

```

Figure 15 CICSJB service class definition

Figure 15 shows the definition for CICSJB, which is a response time with percentile goal where 80% of the transactions need to finish within 0.030 of a second.

**Note:** Data for this service is also accumulated in a workload called CICSCPU. A *workload* is an arbitrary name to group service classes together for reporting or accounting. Transactions that run in this service class have an importance of 2, which is the relative importance of this service class compared to others. The range is 1 - 5, and 1 is the highest importance. Also, discretionary is available, which means that it gets service only if no other service class needs service at that time.

Figure 16 shows the definition for JESTORS.

```

Service Class Name . . . . . : JESTORS
Description . . . . . : CICS JES service class
Workload Name . . . . . : CICSCPU (name or ?)
Base Resource Group . . . . . : _____ (name or ?)
Cpu Critical . . . . . : YES (YES or NO)
I/O Priority Group . . . . . : NORMAL (NORMAL or HIGH)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

      -- Period -- ----- Goal -----
Action # Duration Imp. Description
-----
  1      1 Execution velocity of 70

```

Figure 16 JESTORS service class definition

Figure 16 shows the definition for the JES subsystem service class JESTORS. This is a special case where although all transactions entering the TORs via the network are classified into service class CICSJB and the TOR regions are managed with the other regions that support that service class to achieve that response time goal of 30 milliseconds, the TORs are also managed to their velocity goal because of being managed by BOTH.

**Note:** In this case, the importance of the JESTORS is higher than the CICSJB service class and JESTORS is specified as CPU Critical. CPU Critical means that even if this service class is achieving its goal, lower importance service classes are not allowed to take its CPU when it is executing.

The data for this service class is also accumulated into workload CICSCPU, which is the same workload as the CICS service class.

Figure 17 shows the CICSBTCH service class definition.

```

Service Class Name . . . . . : CICSBTCH
Description . . . . . : batch service class cics
Workload Name . . . . . : CICSCPU (name or ?)
Base Resource Group . . . . . : _____ (name or ?)
Cpu Critical . . . . . : NO (YES or NO)
I/O Priority Group . . . . . : NORMAL (NORMAL or HIGH)

Specify BASE GOAL information. Action Codes: I=Insert new period,
E=Edit period, D=Delete period.

      -- Period -- ----- Goal -----
Action # Duration Imp. Description
-----
  1      1      1      Execution velocity of 70

```

Figure 17 CICSBTCH service class definition

Figure 17 shows the definition for JES subsystem service class CICSBTCH. Although no regions are actually managed to this service class because all of the regions are TRANSACTION-managed, the CPU times, I/O, and real storage are still reported and accumulated into the same workload, CICSCPU.

The RMF report shows the data from all seven CICS regions in workload CICSCPU, including data from both the JES and CICS service classes. An example RMF report is shown in “Analyzing the data” on page 18.

## Verifying the setup

Figure 18 shows the TSO System Display and Search Facility (SDSF) Display Active (DA) display for these regions.

COMMAND INPUT ==>										SCROLL ==> CSR	
NP	JOBNAME	ame	SPag	SCPU%	Workload	SrvClass	SP	ResGroup	Server	Quiesce	ECPU-Time
	CICSA003		0	32	CICSCPU	CICSBTCH	1		YES		8.54
	CICST002		0	32	CICSCPU	JESTORS	1		REG-SERV		24.49
	CICSA004		0	32	CICSCPU	CICSBTCH	1		YES		8.43
	CICST001		0	32	CICSCPU	JESTORS	1		REG-SERV		24.86
	CICSA001		0	32	CICSCPU	CICSBTCH	1		YES		8.19
	CICS2A07		0	32	CICSCPU	CICSBTCH	1		YES		21.53
	CICSA002		0	32	CICSCPU	CICSBTCH	1		YES		8.66

Figure 18 A TSO S.DA display for the CICS regions



In Figure 18 on page 16, when you use TSO SDSF view, all CICS regions appear to have the JES subsystem service classes associated with them. However, under the column Server, in the AORs and the FOR, YES is displayed. YES indicates that these regions, although shown as in service class CICSBTCH, are actually managed by the service classes of the transaction response times running in them. The TORs show service class JESTORs, but under the Server column, REG-SERV is displayed. REG-SERV is the indication that BOTH was specified in the JES subsystem classification rule and that they are managed to a velocity and a response time goal.

## Executing the benchmark

This section describes the execution of the benchmark and how the work was delivered to CICS.

### Repeatable measurements

The first part of this paper describes the issue of repeatable and consistent measurements. The LPAR that is used to support the CICS regions in this benchmark has three dedicated CPs. If you refer to Figure 1 on page 2, you can see that even though our CPs are dedicated, the L3 and L4 caches are shared with other CPs that are used by other LPARs. So, this situation is not perfect; it can lead to CPU variation because those caches can have their data invalidated by those CPs that are used by the other LPARs. It is important to understand this situation before we can continue to show how the data can be collected and then analyzed.

We ran this benchmark by using IBM Workload Simulator for z/OS (WSim) running on one LPAR, simulating 10000 LU2 clients in session with the TORs. The system under test was on a separate LPAR in the same sysplex. All network traffic was routed by way of a coupling facility from one LPAR to the other. A “user think time” was adjusted to achieve the required transaction rate in the CICS regions, and the workload was allowed to run for a period to ensure that all programs were loaded and the LSR pools were populated. After the stabilization period was complete, the performance data collection was started.

### Collecting the data

No specific changes to any default CICS parameters were needed to support the data that was collected during this benchmark. Data was collected for a 5-minute period, which is relatively short but adequate for the demonstration in this paper.

RMF, CICS Performance Class Monitoring, CICS statistics, and HIS were all synchronized and started and ended together. We use an automation tool that enters commands on the IBM MVST<sup>™</sup> console on a time-based interval. Clients can use the tool that is most convenient for them for this role.

To generate the RMF interval, we chose to simply start and stop RMF at the appropriate times, which creates an interval report for that period rather than trying to synchronize on a time basis.

When the workload was running at its stabilized state, the CICS statistics were reset by using the commands that are shown in Figure 4 on page 6. CICS Performance Class Monitoring was turned on by using the commands that are shown in Figure 5 on page 6. Example 1 shows these commands.

---

*Example 1 Reset command and command to start CICS Performance Class Monitoring*

---

RMF Monitor I was started using  
S RMF.R and then Monitor III was started using  
F R,START III

---

HIS was also started for counters only. For more information about setting HIS up, see *Setting Up and Using the IBM System z CPU Measurement Facility with z/OS*, REDP-4727:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4727.pdf>

After 5 minutes elapsed, RMF and HIS were stopped, a command to perform CICS statistics was issued, as shown in Figure 4 on page 6, and CICS Monitoring was stopped, as shown in Figure 5 on page 6. The SMF dataset was copied for later processing.

## Analyzing the data

This section describes the post-processing and analysis of the data that was collected during the benchmark.

### RMF Monitor I data

Figure 19 shows the RMF Monitor I post-processing.

```
//POST      EXEC PGM=ERBRMFPP,REGION=0M
//MFR00001 DD SYSOUT=*
//PPXSRPTS DD SYSOUT=*
//MFPINPUT DD DSN=SMF.COPY,DISP=SHR
//SYSOUT    DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//MFPMMSGDS DD SYSOUT=A
//SYSIN     DD *
REPORTS (ALL)
REPORTS (ESS)
SYSRPTS (WLMGL (POLICY, WGP, RCLASS, SC, WGROUP) )
/*
```

Figure 19 RMF Monitor I post-processing

Figure 19 shows a window of the JCL for the RMF post-processing step that was used to produce the RMF Monitor I report.

The following figures are extracts from the RMF Monitor I report. Using a top-down approach, we look at the LPAR configuration first to see how busy it is.

Figure 20 on page 19 shows the RMF I CPU Activity report.

z/OS V2R1				SYSTEM ID MV2A		DATE 10/20/2014	
				RPT VERSION V2R1 RMF		TIME 19.03.36	
CPU	2827	CPC CAPACITY	9066	SEQUENCE CODE 0000000000058D27			
MODEL	799	CHANGE REASON=NONE		HIPERDISPATCH=YES			
H/W MODEL	HA1						
---CPU---		TIME %				LOG PROC	--I/O INTERRUPTS--
NUM	TYPE	ONLINE	LPAR BUSY	MVS BUSY	PARKED	SHARE %	RATE % VIA TPI
0	CP	100.00	30.50	30.50	0.00	100.0	1735 0.29
1	CP	100.00	27.71	27.71	0.00	100.0	1784 0.27
2	CP	100.00	27.59	27.59	0.00	100.0	1839 0.26
TOTAL/AVERAGE			28.60	28.60		300.0	5358 0.27

Figure 20 RMF I CPU Activity report

The RMF I CPU Activity report shows three online CPs, which on average are 28.6% busy for this 5-minute interval.

Figure 21 shows the RMF I Partition report.

MVS PARTITION NAME				MVS2A		NUMBER OF PHYSICAL PROCESSORS				101		
IMAGE CAPACITY				2289		CP				99		
NUMBER OF CONFIGURED PARTITIONS				57		AAP				1		
WAIT COMPLETION				NO		IIP				1		
DISPATCH INTERVAL				DYNAMIC								
----- PARTITION DATA -----						-- LOGICAL PARTITION PROCESSOR DATA --				-- AVERAGE PROCESSOR		
-----MSU----- -CAPPING--						PROCESSOR-		-----DISPATCH TIME DATA----		LOGICAL PROCESSORS		
NAME	S	WGT	DEF	ACT	DEF	WLM%	NUM	TYPE	EFFECTIVE	TOTAL	EFFECTIVE	TOTAL
MVS2A	A	DED	0	275		0.0	3	CP	00.15.08.632	00.15.08.708	99.99	100.0
AACF01	A	DED	0	366		0.0	4	CP	00.20.11.608	00.20.11.615	100.0	100.0
MVSAA	A	DED	0	916		0.0	10	CP	00.50.28.958	00.50.29.015	100.0	100.0
MVSAB	A	DED	0	275		0.0	3	CP	00.15.08.654	00.15.08.703	99.99	100.0
MVSAC	A	DED	0	916		0.0	10	CP	00.50.28.955	00.50.29.013	100.0	100.0

Figure 22 on page 19 shows the report class for the two TORs. Some of the data is populated from the JES classification and some of the data is populated from the CICS classification rules. Remember that both the JES and the CICS classification rules pointed to the same report class name CICSTORS.

The report shows two address spaces that are executing in this report, Class (AVG) and on average, which use 15.65% of a CP. The report shows 1322.7 transactions per second executing. In total, 400653 transactions were executed across the two address spaces and 365760 transactions were routed to the AORs for execution. ACTUAL 16 represents the average response time of 16 milliseconds for all transactions. EXECUTION represents the response time of 17 milliseconds in the AORs for those transactions that were routed. Logically, you expect the EXECUTION to be lower than the ACTUAL because it is a subset of the response time of routed transactions. However, in this case, 34893 (ended - executed) transactions were run locally and had short response times, therefore reducing the ACTUAL times. These 34893 transactions were CICS system transactions that were initiated and not routed. Only the user transactions were dynamically routed.

The total and average usage of real storage is also displayed.

## AOR Report class

WLM was configured so that the four AORs were reported in the CICSAORS Report class report (Figure 23).

REPORT BY: POLICY=POLICY				REPORT CLASS=CICSAORS											
				DESCRIPTION =Report for WLM AORs - Johnb											
-TRANSACTIONS-	TRANS-TIME	HHH.MM.SS.TTT	--DASD I/O--	---	SERVICE---	SERVICE TIME	---	APPL %---	---	PROMOTED---	---	STORAGE----			
AVG	4.00	ACTUAL	12	SSCHRT	519.4	IOC	78703	CPU	106.207	CP	35.66	BLK	0.000	AVG	34937.75
MPL	4.00	EXECUTION	0	RESP	0.0	CPU	7586K	SRB	1.494	AAPCP	0.00	ENQ	0.000	TOTAL	139755.4
ENDED	31210	QUEUED	0	CONN	0.0	MSQ	0	RCT	0.000	IIPCP	0.00	CRM	0.000	SHARED	0.00
END/S	103.04	R/S AFFIN	0	DISC	0.0	SRB	106682	IIT	0.309			LCK	0.144		
#SWAPS	0	INELIGIBLE	0	Q+PEND	0.0	TOT	7772K	HST	0.000	AAP	N/A	SUP	0.000	-PAGE-IN RATES-	
EXCTD	0	CONVERSION	0	IOSQ	0.0	/SEC	25657	AAP	N/A	IIP	0.00			SINGLE	0.0
AVG ENC	0.00	STD DEV	47					IIP	0.000					BLOCK	0.0
REM ENC	0.00					ABSRPTN	6414							SHARED	0.0
MS ENC	0.00					TRX SERV	6414							HSP	0.0

Figure 23 Report class for AORs

Figure 23 is an extract of the report class for the four AORs. The transaction rate reported in this report is relatively low compared to the actual number executing in these AORs because the transactions that were routed to these AOR regions from the TORs are not reported by RMF. Their counts are accumulated back in the TOR Report class. Only transactions that are actually initiated in these AOR regions are counted in this report. The 31210 transactions counted in this report are a result of some of the routed transactions issuing an EXEC CICS START of a new transaction, which results in a new classification in the AOR and counts toward ENDED transactions for this report class.

## FOR Report class

WLM was configured so that the FOR was reported in the CICSFOR Report class report (Figure 24 on page 21).

REPORT BY: POLICY=POLICY				REPORT CLASS=CICSFOR												
				DESCRIPTION =												
-TRANSACTIONS-	TRANS-TIME	HHH.MM.SS.TTT	--DASD	I/O--	---	SERVICE---	SERVICE	TIME	---	APPL	%---	--PROMOTED--	----	STORAGE----		
AVG	1.00	ACTUAL	95	SSCHRT	3801	IOC	575942	CPU	50.856	CP	18.41	BLK	0.000	AVG	23342.04	
MPL	1.00	EXECUTION	0	RESP	0.1	CPU	3633K	SRB	2.655	AAPCP	0.00	ENQ	0.000	TOTAL	23342.74	
ENDED	401	QUEUED	0	CONN	0.0	MSO	0	RCT	0.000	IIPCP	0.00	CRM	0.000	SHARED	1.00	
END/S	1.32	R/S AFFIN	0	DISC	0.0	SRB	189670	IIT	2.250			LCK	1.760			
#SWAPS	0	INELIGIBLE	0	Q+PEND	0.0	TOT	4398K	HST	0.000	AAP	N/A	SUP	0.000	-PAGE-IN	RATES-	
EXCTD	0	CONVERSION	0	IOSQ	0.0	/SEC	14520	AAP	N/A	IIP	0.00			SINGLE	0.0	
AVG ENC	0.00	STD DEV	119					IIP	0.000					BLOCK	0.0	
REM ENC	0.00					ABSRPTN	15K							SHARED	0.0	
MS ENC	0.00					TRX SERV	15K							HSP	0.0	

Figure 24 Report class for FOR

Figure 24 shows the transactions that are reported by WLM for the FOR. Because this report is a pure FOR and nearly all of the transactions are CSMLs, we do not expect to see many ENDED transactions in this report.

Looking at the CICS statistics for the FOR, which we show later, we see that actually 339527 CSML transactions were executed, which are not counted individually by WLM. Also, 286 CSKP and 117 CSSY transactions were executed, for a total of 403 transactions that are classified in this report and counted in the 5-minute interval. This minimal difference between the RMF count of 401 and the statistics is due to timing.

## WLM workload reporting

WLM was also configured so that all of the service classes that were used were grouped into one workload for reporting (Figure 25).

REPORT BY: POLICY=POLICY				WORKLOAD=CICSCPU											
Total for all benchmark CICS Regions															
-TRANSACTIONS-	TRANS-TIME	HHH.MM.SS.TTT	--DASD	I/O--	---	SERVICE---	SERVICE	TIME	---	APPL	%---	--PROMOTED--	----	STORAGE----	
AVG	7.00	ACTUAL	16	SSCHRT	4320	IOC	654645	CPU	194.143	CP	69.72	BLK	0.000	AVG	33794.87
MPL	7.00	EXECUTION	17	RESP	0.1	CPU	13867K	SRB	14.477	AAPCP	0.00	ENQ	0.000	TOTAL	236571.5
ENDED	432264	QUEUED	0	CONN	0.0	MSO	0	RCT	0.000	IIPCP	0.00	CRM	0.000	SHARED	2.00
END/S	1427.06	R/S AFFIN	0	DISC	0.0	SRB	1034K	IIT	2.559			LCK	1.960		
#SWAPS	0	INELIGIBLE	0	Q+PEND	0.0	TOT	15556K	HST	0.000	AAP	N/A	SUP	0.000	-PAGE-IN	RATES-
EXCTD	365760	CONVERSION	0	IOSQ	0.0	/SEC	51356	AAP	N/A	IIP	0.00			SINGLE	0.0
AVG ENC	0.00	STD DEV	30					IIP	0.000					BLOCK	0.0
REM ENC	0.00					ABSRPTN	7336							SHARED	0.0
MS ENC	0.00					TRX SERV	7336							HSP	0.0

Figure 25 Workload report for all CICS regions

Figure 25 shows the report for workload CICS CPU. Seven CICS regions exist in total: two TORs, four AORs, and one FOR. In total, they used 69.72% of one CP. The transaction rate of 1427.06 consists of all transactions that started in the TORs, transactions that were initiated in the AORs, and CICS system-type transactions that were initiated within the FOR.

The data from JESTORS, CICSBTCH, and CICSJB is accumulated in this report.

To calculate an average CPU per transaction for comparison, divide the 0.6972 of a CP by 1427.06 = 0.000488 seconds/transaction.

The total real storage that was used was 33794.87 frames. A frame is 4096 bytes so 33794.87 x 4096 = 138.4 M.

## CICS service class reporting

Figure 26 on page 22 shows the CICS service class report.

REPORT BY: POLICY=POLICY		WORKLOAD=CICSCPU	SERVICE CLASS=CICSJB	RESOURCE GROUP=*NONE	PERIOD=1	IMPORTANCE=2
			CRITICAL =NONE			
-TRANSACTIONS- TRANS-TIME HHH.MM.SS.TTT						
AVG	0.00	ACTUAL	16			
MPL	0.00	EXECUTION	17			
ENDED	432264	QUEUED	0			
END/S	1427.06	R/S AFFIN	0			
#SWAPS	0	INELIGIBLE	0			
EXCTD	365760	CONVERSION	0			
AVG ENC	0.00	STD DEV	30			
REM ENC	0.00					
MS ENC	0.00					
----- STATE SAMPLES BREAKDOWN (%) -----						
SUB	P	TIME	RESP	ACTIVE	READY	IDLE
TYPE		(%)	SUB APPL	MISC	CONV	TIME LOCK I/O
CICS	BTE	655	0.0 0.0	0.0	21.4	54.6 14.6 9.4 0.0 0.0
CICS	EXE	94.8	0.1 0.0	0.4	4.5	90.4 0.0 0.0 3.2 1.4
-----STATE-----						
SWITCHED SAMPL (%)						
LOCAL SYSPL REMOT						
14.6 0.0 0.0						
96.8 0.0 0.0						

Figure 26 CICS service class

Figure 26 shows the service class CICSJB. The report has four parts. The first part shows the transaction rates and response times that we saw in the Workload report. The second part shows the breakdown of the WLM samples that were taken. This second part is not always useful because long-running tasks can distort the numbers. The third part shows the Performance Index (PI). A PI of 1 means that the goal is achieved. A PI of less than 1 means that the goal is overachieved. A PI that is greater than 1 means that the goal is not being achieved. The fourth part of the report shows the distribution by response time. In this case, the first bucket indicates that 65.5% of the transactions complete within 0.015 seconds.

## RMF Monitor III data

RMF Monitor III data was post-processed by using the JCL that is shown in Figure 27.

```
//STEP1 EXEC PGM=IFASMFDP
//DUMPIN DD DSN=JOHNB.MEASURE.SMFCOPY,DISP=SHR
//DUMPOUT DD DSN=&&SMF,UNIT=SYSDA,
// DISP=(NEW,PASS),SPACE=(CYL,(100,100))
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
INDD (DUMPIN,OPTIONS (DUMP))
OUTDD (DUMPOUT,TYPE (070:080))
//POST EXEC PGM=ERBRMFPP,REGION=0M
//MFPINPUT DD DSN=&&SMF,DISP=(OLD,PASS)
//SYSUDUMP DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//MFPMSGDS DD SYSOUT=A
//SYSIN DD *
NOSUMMARY
SYSRPTS (CF)
SYSOUT (A)
REPORTS (XCF)
/*
```

Figure 27 RMF Monitor III post-processing

This benchmark used the coupling facility (CF) for CICS log structures and CICS temporary storage. Both the access rates and response time are recorded by RMF III. If a complete sysplex-wide view is required, the SMF data from all LPARs in the sysplex must be sorted and concatenated as input to the first step.

Figure 28 show extracts from the report.

STRUCTURE NAME = LOGS_DSW TYPE = LIST STATUS = ACTIVE												
SYSTEM NAME	# REQ	REQUESTS				DELAYED REQUESTS						
	TOTAL	#	% OF	-SERV	TIME (MIC) -	REASON	#	% OF	----	AVG TIME (MIC)	-----	
	AVG/SEC	REQ	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	
MV2A	38436	SYNC	38K	98.6	9.2	3.0	NO SCH	0	0.0	0.0	0.0	0.0
	129.4	ASYN	540	1.4	173.6	204.4	PR WT	0	0.0	0.0	0.0	0.0
		CHNGD	0	0.0	INCLUDED IN ASYN		PR CMP	0	0.0	0.0	0.0	0.0
		SUPPR	0	0.0			DUMP	0	0.0	0.0	0.0	0.0

Figure 28 CICS CF log structure

Figure 28 shows the statistics for the CF structure that is associated with the CICS log streams. In this benchmark, only the FOR writes any log records so this value of 129.4 accesses per second can be tied back to the CICS log stream statistics that were collected for the FOR. Most requests are processed synchronously. Synchronous requests have quicker response times but they can be altered to asynchronous by MVS, based on monitoring the past performance of similar requests. Synchronous requests can also be changed to asynchronous due to the lack of a resource, such as a subchannel, and this situation is counted on the CHNDG row. No changed requests show on this report. Most of the requests have a response time of 9.2 microseconds.

Figure 29 shows the CICS temporary storage structure.

STRUCTURE NAME = DFHXQLS_TSP00L1 TYPE = LIST STATUS = ACTIVE												
SYSTEM NAME	# REQ	REQUESTS				DELAYED REQUESTS						
	TOTAL	#	% OF	-SERV	TIME (MIC) -	REASON	#	% OF	----	AVG TIME (MIC)	-----	
	AVG/SEC	REQ	ALL	AVG	STD_DEV		REQ	REQ	/DEL	STD_DEV	/ALL	
MV2A	648K	SYNC	648K	100	3.9	1.6	NO SCH	0	0.0	0.0	0.0	0.0
	2183	ASYN	0	0.0	0.0	0.0	PR WT	0	0.0	0.0	0.0	0.0
		CHNGD	0	0.0	INCLUDED IN ASYN		PR CMP	0	0.0	0.0	0.0	0.0
		SUPPR	0	0.0			DUMP	0	0.0	0.0	0.0	0.0

Figure 29 CICS temporary storage structure

Figure 29 shows the RMF Monitor III statistics for the CICS temporary storage server coupling facility structure. The report shows 2183 synchronous CICS temporary storage requests per second with a 3.9-microsecond response time.

Figure 30 shows the coupling facility processor summary.

PROCESSOR SUMMARY							
-----							
COUPLING FACILITY	2827	MODEL	HA1	CFLEVEL	19	DYNDISP	OFF
AVERAGE CF UTILIZATION	(% BUSY)	0.5	LOGICAL PROCESSORS:	DEFINED	2	EFFECTIVE	2.0
				SHARED	0	AVG WEIGHT	0.0

Figure 30 Coupling facility processor summary

In Figure 30 on page 23, the RMF Monitor III coupling facility summary shows the coupling facility with two dedicated CPs and 0.5% busy. Dynamic Dispatch (DYNDISP), as shown, is turned off, which means that the frequency of polling that is performed by a coupling facility to look for work does not slow down when the CF becomes less busy. Turning on DYNDISP can increase the response times for infrequent requests.

## CICS TS statistics

The CICS statistics were processed by using the CICS statistics utility program, DFHSTUP. Figure 31 shows an example of the JCL that extracted the statistics for the FOR.

```
//STUP      EXEC PGM=DFHSTUP
//STEPLIB DD   DSN=CTS520.CICS690.SDFHLOAD,DISP=SHR
//DFHSTATS DD   DSN=SMF.COPY,DISP=SHR
//DFHSTWRK DD   UNIT=SYSDA,SPACE=(CYL,(6))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(6))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(6))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(6))
//SORTWK04 DD   UNIT=SYSDA,SPACE=(CYL,(6))
//SORTWK05 DD   UNIT=SYSDA,SPACE=(CYL,(6))
//DFHPRINT DD   SYSOUT=A
//SYSOUT DD     SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSIN DD *
      SELECT APPLID=(IYCUZC07)
      COLLECTION TYPE=REQUESTED
```

Figure 31 CICS statistics utility

Figure 31 shows the post-processing for only the FOR in this configuration.

**Note:** TYPE=REQUESTED was specified in Figure 31 because a **CEMT PERFORM STATISTICS** was used to write the statistics to the SMF dataset during this benchmark.

By using the FOR as an example, the following extracts from the statistics report show several of the key indicators for CPU usage (Figure 32).

DISPATCHER STATISTICS	
Dispatcher Start Date and Time. . . . .	10/20/2014 18:11:08.2425
Address Space CPU Time. . . . .	00:00:50.984505
Address Space SRB Time. . . . .	00:00:02.663478
Current number of dispatcher tasks. . . . .	40
Peak number of dispatcher tasks. . . . .	88
Current ICV time (msec). . . . .	8000
Current ICVR time (msec). . . . .	5000
Current ICVTSD time (msec). . . . .	500
Current PRTYAGE time (msec). . . . .	32768

Figure 32 CICS Dispatcher Statistics



Figure 32 on page 24 shows an extract of the CICS Dispatcher Statistics for the FOR. Although not captured in the window, there was a period of 304 seconds since the last reset. The CPU usage for the address space can be calculated from this report as  $TCB + SRB / \text{Interval}$  or  $50.98+2.66 / 304 = 17.5\%$  of 1 CP, which relates back to the RMF report class for the FOR in Figure 24 on page 21.

A small difference exists because RMF shows 18.41% of one CP. However, the CICS statistics show only task control block (TCB) and service request block (SRB) CPU usage compared to the RMF that shows TCB, SRB, and in this case, IIT. IIT is the time that is used for processing I/O interrupts. Excluding the IIT, both the RMF and the CICS statistics show the same percentage value. Generally, the times that are reported in the CICS statistics match the times that are reported by RMF.

Figure 33 shows the CICS TCB Mode Statistics.

CICS TCB Mode Statistics										
TCB Mode	Open	TCB Pool	< TCBs Attached >		<- TCBs In Use ->		TCB Attaches	<- Dispatchable Queue ->		
			Current	Peak	Current	Peak		Current	Peak	Average
QR	No	N/A	1	1	1	1	0	1	49	1.90

Figure 33 CICS TCB Mode Statistics

Figure 33 shows the TCB statistics for only the QR TCB. Because this is a pure FOR with no threadsafe applications, only the QR TCB is used. The report shows that on average when a task was dispatched, 1.9 queued tasks were also ready to go. A high value in this report is an indicator of contention for the QR TCB.

Figure 34 shows the Requested Statistics Report.

Requested Statistics Report										Collection Date-Time 10/20/2014-19:08:39	Last Reset 19:03:35	Applid IYCUZC07	Jobname
TCB Mode	< TCBs Attached >		TCB Attaches	Attach Failures	MVS Waits	Accumulated Time in MVS wait	Accumulated Time Dispatched	Accumulated Time / TCB					
	Current	Peak											
QR	1	1	0	0	1464998	00:04:09.657870	00:00:54.281921	00:00:50.984259					
RO	1	1	0	0	0	00:00:00.000000	00:00:00.000000	00:00:00.000000					
CO	0	0	0	0	0	00:00:00.000000	00:00:00.000000	00:00:00.000000					

Figure 34 CICS Requested Statistics Report

Figure 34 shows the QR TCB usage. In this case, all of the CPU is accumulated on the QR TCB so the accumulated TCB time that is shown in this report matches the time that is shown in the address CPU time in Figure 32 on page 24. The dispatch time compared to CPU time provides an indication of any CPU constraint. If the QR dispatch time is much higher than the CPU time on the QR TCB, after an interrupt, other address spaces are dispatched before this address space. The ratio  $50.984259/54.281921$  gives a CPU to dispatch time ratio of 92%, which is good. The use of this ratio for an indication of CPU constraint is only valid for the QR TCB, not for open TCBs or other TCBs that might give up the CPU without using a CICS dispatcher wait.

Figure 35 on page 26 shows CICS Monitoring global statistics.

MVS WLM Mode. . . . .	: Goal	CEC Machine Type and Model Number . . .	: 2827-799
MVS WLM Server. . . . .	: Yes		
MVS WLM Workload Name . . . . .	: CICSCPU		
MVS WLM Service Class . . . . .	: CICSBTCH		
MVS WLM Report Class. . . . .	: CICSFORS		
MVS WLM Resource Group. . . . .	:		
WLM Manage Regions Using Goals of . . .	: Transaction		
MVS WLM Goal Type . . . . .	: Velocity		
MVS WLM Goal Value. . . . .	: 70		
MVS WLM Goal Importance . . . . .	: 1		
MVS WLM CPU Critical. . . . .	: No		
MVS WLM Storage Critical. . . . .	: No		
User transactions ended . . . . .	: 339520		
System transactions ended . . . . .	: 403		
Time last user transaction attached . . .	: 10/20/2014 19:08:39.6298	MXT at last user transaction attach . . .	: 200
Time last user transaction ended. . . .	: 10/20/2014 19:08:39.6297	Current tasks at last attach. . . . .	: 19
Average user transaction resp time. . .	: 00:00:00.019857		
Peak user transaction resp time . . . .	: 00:00:00.280093		
Peak user transaction resp time at. . .	: 10/20/2014 19:07:53.5370		

Figure 35 CICS Monitoring global statistics

Figure 35 shows an extract from the CICS Monitoring global statistics. This part shows how the CICS region is managed by WLM. It shows that the service class that is associated with the FOR is CICSBTCH. However, the window also shows that the region is not being managed to that because it shows that the region is being managed by using the goals of Transaction. So, this region is managed by the service classes of the transactions that are running in it. This is a pure FOR so the CSMI mirror tasks carry the service class token from the TOR where the originating transaction was classified, in this case, CICSJB. The transaction's Performance Class Monitoring record for the region where the transaction was classified contains the service class name in field SRVCLASS. If the transaction is routed elsewhere, the Performance Class Monitoring record for the transaction in the target region does not contain the service class name. The only way to see the service classes that are supported by a region is by running a CICS Performance Analysis (PA) Workload Activity report or by looking at the WLM definitions.

The CICS Monitoring global statistics report also provides information about current tasks and average response times.

When you compare benchmarks, it is good to check that the number of file requests and the types of requests are similar to a previous benchmark and that the same number of I/Os, and so on, completed. The I/O rate is also shown in the RMF Monitor I Workload report, as shown in Figure 24 on page 21, where the DASD I/O column shows that 3801 subchannels started per second (SSCHRT) on average. When multiplied by the 304-second interval, the total is 1155504, which is similar to the number of EXCPs that are reported in the file statistics if you add the Index and Data EXCPs together.

Figure 36 shows the CICS file statistics report.

FILES - Requests Information											
File Name	Get Requests	Get Upd Requests	Browse Requests	Update Requests	Add Requests	Delete Requests	Brws Upd Requests	VSAM Data	EXCP Index	Requests RLS	req Timeouts
*TOTALS*	3037247	440371	0	401447	301698	38915	0	1149651	5310	0	0

Figure 36 CICS file statistics

The CICS statistics for the FOR region has more information in it than we can show, so this paper covers only some of the key indicators of CPU usage.

## CICS Performance Class Monitoring records

For the 5-minute measurement interval that CICS Performance Monitoring was turned on and collected data, as shown in Figure 5 on page 6, CICS Performance Analyzer for z/OS (CICS PA) was used to post-process these records.

**Important:** We do not recommend using CICS Performance Class Monitoring records for CPU comparisons between benchmarks.

The following reports are included to show examples of how CICS PA can be used to give you valuable insight into the characteristics of CICS regions and how CICS PA can be used to diagnose performance problems or for CPU chargeback.

For more information about this product and how to produce the following reports, see this website:

<http://www-03.ibm.com/software/products/en/cics-panaly>

The following windows show several reports that were generated by using CICS PA.

Figure 37 shows the CICS Performance Analyzer Performance Summary report.

V5R2M0		CICS Performance Analyzer Performance Summary									
SUMM0001 Printed at 11:47:12 10/23/2014					Data from 19:03:36 10/20/2014 to 19:08:41 10/20/2014						
Start Interval	APPLID	#Tasks	Avg User CPU Time	Total User CPU Time	Total LogWrite Count	Avg Response Time	Avg IR Wait Count	Avg FC Total Count	Total TS Count	Total TD Count	Total Total
19:00:00	IYCUA001	100068	.000233	23.27531	0	.017939	11	0	14310	31267	
19:00:00	IYCUA002	99186	.000236	23.39410	0	.017484	11	0	14805	30683	
19:00:00	IYCUA003	100097	.000232	23.19465	0	.017949	11	0	14520	30727	
19:00:00	IYCUA004	100306	.000235	23.55314	0	.017909	11	0	14575	31560	
19:00:00	IYCUT001	199531	.000063	12.62955	0	.017077	1	0	34497	1	
19:00:00	IYCUT002	203847	.000062	12.69488	0	.017622	1	0	35361	1	
19:00:00	IYCUZC07	341167	.000119	40.61197	35983	.019966	12	12	0	575	
19:00:00		1144202	.000139	159.3536	35983	.018295	8	3	128068	124814	
Total		1144202	.000139	159.3536	35983	.018295	8	3	128068	124814	

Figure 37 CICS PA Summary report for all address spaces

Figure 37 is a summary of all transactions, sorted by APPLID, that were executed. Only a subset of the CICS Monitoring fields is included in the report.

The CPU time that is reported by CICS Performance Class Monitoring is reported at a transaction level; it does not include the CPU time that is used, for example, by the CICS dispatcher to get the transaction running or all of the CPU that is used to get the transaction redispached after a wait. It also does not include SRB or CPU time that is used on any TCBs that are not CICS. For example, the CICS Performance Analyzer Performance Summary report shows the Total CPU time for all transactions across all regions as 159.3536 seconds in this 305-second interval.

The RMF report that is shown in Figure 25 on page 21 shows the total TCB CPU time as 194.143 seconds. So, CICS Monitoring shows about 81% of the TCB CPU time that is used.

**Important:** Be aware of this difference when you compare benchmarks by using only monitoring records. It is possible that one version of the software that is used in the comparison has changes that will improve parts of the code that are not accounted for by monitoring. It is always best to look at the total CPU times from the CICS statistics or RMF.

Looking at an example for the FOR only, Table 1 shows a comparison of CPU time.

Table 1 CPU time comparison for the FOR

	CICS statistics	RMF Monitor I	CICS Monitoring
TCB time	50.98	50.85	40.61
SRB time	2.66	2.65	N/A
IIT time	N/A	2.25	N/A

The following windows show a CICS PA summary report for only one of the AORs. It sorted and listed the summary of transactions by the highest average CPU per transaction first.

Figure 38 shows the CICS summary report for the top CPU users.

Tran	APPLID	Start	Avg User CPU	Total User CPU	#Tasks	Avg QR CPU	Avg Response	Avg DSCHMDLY	Avg IR Wait	Avg DB2 Reqs
		Interval	Time	Time		Time	Time	Count	Count	Count
CEMT	IYCUA001	19:00:00	.010158	.010158	1	.010158	.030154	0	0	0
OE5	IYCUA001	19:00:00	.000511	1.244797	2435	.000511	.101760	0	22	0
OX5	IYCUA001	19:00:00	.000511	1.218203	2383	.000511	.102719	0	22	0
PS3	IYCUA001	19:00:00	.000448	1.946039	4339	.000448	.002614	0	34	0
PX3	IYCUA001	19:00:00	.000448	1.989190	4440	.000448	.002627	0	34	0
SC2	IYCUA001	19:00:00	.000306	.452160	1479	.000306	.088682	0	18	0
SX2	IYCUA001	19:00:00	.000305	.452756	1484	.000305	.088476	0	18	0
PX2	IYCUA001	19:00:00	.000300	1.460865	4862	.000300	.002200	0	18	0
PS2	IYCUA001	19:00:00	.000300	1.475239	4911	.000300	.002198	0	18	0

Figure 38 CICS PA summary report for the top CPU users

The report in Figure 38 shows OE5 as the user transaction with the highest average CPU time per transaction, although OE5 is not the highest user of CPU time in total. To look at the resources that this transaction was accessing and waiting on, a CICS PA Wait analysis report provides an overall view.

Figure 39 shows the CICS PA Wait analysis report.

Tran=OE5										
Summary Data			----- Time -----		----- Count -----		----- Ratio -----			
			Total	Average	Total	Average				
# Tasks					2435					
Response Time			247.785715	0.101760						
Dispatch Time			1.343401	0.000552	58443	24.0	0.5% of Response			
CPU Time			1.244797	0.000511	58443	24.0	92.7% of Dispatch			
Suspend Wait Time			246.442345	0.101208	58443	24.0	99.5% of Response			
Dispatch Wait Time			1.436677	0.000590	56008	23.0	0.6% of Suspend			
QR TCB Redispach Wait Time			1.436677	0.000590	56008	23.0	100.0% of Dispwait			
Resource Manager Interface (RMI) elapsed time			0.000602	0.000000	4870	2.0	0.0% of Response			
Resource Manager Interface (RMI) suspend time			0.000000	0.000000	0	0.0	0.0% of Suspend			
Suspend Detail										
			----- Suspend Time -----				----- Count -----			
			Total	Average	%age	Graph	Total	Average		
IRIOWTT MRO link wait time			246.325038	0.101160	100.0%	*****	56003	23.0		
DSPDELAY First dispatch wait time			0.114830	0.000047	0.0%		2435	1.0		
TSSHWAIT Asynchronous Shared TS wait time			0.002477	0.000001	0.0%		5	0.0		

Figure 39 CICS PA Wait analysis report

Figure 39 on page 28 shows a CICS PA Wait analysis report for the OE5 transaction. This report is based on only one of the AORs for all OE5s running in that region. The selection can be either for a single instance of a transaction if a transaction was identified as having poor performance or as in this case, a summary of them all. This report shows that all of the wait time that was experienced for these transactions was in IRIOWTT and that the 0.101160 seconds were accumulated from an average of 23 multiregion operation (MRO) link waits.

To look closely at the flows in the OE5 transaction, you can run a CICS PA Cross System Work report. Figure 40 shows an extract from this report.

Tran	Userid	SC	TranType	Term	LUName	Request Type	Program	Fcty T/Name	Conn Name	NETName	UOW Seq	APPLID	R Task T	Stop Time	Response Time	A B
OE5	JOHNB	TO	U	0000	DX0000	TR:A001		T/0000		GBIBMIYA.DX0000	1	IYCUT001	78563	T 19:05:00.254	.1297	
OE5	JOHNB	TO	U	A2	IYCUT001	AP:	DSW0E5VV	S/0000	T001	GBIBMIYA.DX0000	1	IYCUA001	90888	T 19:05:00.254	.1291	
CSMI	JOHNB	TO	UM	A6	IYCUA001	FS:F---	DFHMIRS	T/A6	A001	GBIBMIYA.DX0000	1	IYCUZC07	69456	T 19:05:00.254	.1287	
----																
OE5	JOHNB	TO	U	0000	DX0000	TR:A001		T/0000		GBIBMIYA.DX0000	1	IYCUT001	36574	T 19:06:28.605	.1109	
OE5	JOHNB	TO	U	A2	IYCUT001	AP:	DSW0E5VV	S/0000	T001	GBIBMIYA.DX0000	1	IYCUA001	20116	T 19:06:28.605	.1097	
CSMI	JOHNB	TO	UM	A6	IYCUA001	FS:F---	DFHMIRS	T/A6	A001	GBIBMIYA.DX0000	1	IYCUZC07	68522	T 19:06:28.605	.1092	
----																
OE5	JOHNB	TO	U	0000	DX0000	TR:A001		T/0000		GBIBMIYA.DX0000	1	IYCUT001	3850	T 19:08:11.122	.0668	
OE5	JOHNB	TO	U	A4	IYCUT001	AP:	DSW0E5VV	S/0000	T001	GBIBMIYA.DX0000	1	IYCUA001	53773	T 19:08:11.122	.0664	
CSMI	JOHNB	TO	UM	A6	IYCUA001	FS:F---	DFHMIRS	T/A6	A001	GBIBMIYA.DX0000	1	IYCUZC07	83214	T 19:08:11.122	.0661	

Figure 40 CICS PA Cross System Work report

Figure 40 shows an extract from a CICS PA Cross System Work report for the OE5 transaction. The report lists each execution of the transaction and shows where the transaction originated, where the transaction went, and any function-shipping that the transaction performed. For example, the first entry shows that it started in region IYCUT001 from terminal DX0000. From there, it was transaction-routed to IYCUA001 where the application program DSW0E5VV function-shipped file requests to IYCUZC07, which were handled by the CSMI mirror task 69456. For more detail about how to understand this report, see this document:

<http://publibfp.dhe.ibm.com/epubs/pdf/cpaabc52.pdf>

For this demonstration only, Figure 41 shows an extract of the CICS PA Performance List report for the mirror task 69556, which supported this particular OE5 task.

CICS Performance Analyzer Performance List												
at 14:13:30 10/24/2014			Data from 19:05:00 10/20/2014					APPLID IYCUZC07		Page	1	
Request Activity - Detail												
TaskNo	Start Time	Stop Time	Response Time	FCADD	FCBROWSE	FCDELETE	FCGET	FCPUT	FC Total	FC Wait Time	LogWrite	JC Wait Count
69456	19:05:00.125	19:05:00.254	.128677	4	0	0	9	9	22	.005883	0	14

Figure 41 CICS PA Performance List report

The fields that are chosen for display in this CICS PA detailed report were the file control accesses only. This report displays a total of 22 file control requests, which consist of nine Reads, nine Rewrites, and four Writes. This ties into the report in Figure 39 on page 28 where the OE5 transaction in the AOR has an average IRIOWTT count of 23: 22 file requests function-shipped and one Syncpoint flow. Also, a LogWrite count of 0 but a JC Wait Count of 14 is interesting, which indicates that the transaction waited 14 times for the log buffer to be written but it was never the one that was charged with the physical log write. That is, each time that this task wanted the log buffer to be written, another task already requested a log force and set the defer log interval running. That original task is the one that is dispatched when the time elapses, and it gets the LogWrite count.

A CICS PA Transaction File usage summary report, as shown in Figure 42, provides more detail about the files that were used in the transaction.

Tran	#Tasks	***** FC Calls *****							***** I/O Waits *****		Excl
		Get	Put	Browse	Add	Delete	Total	File	RLS	CFDT	
CSMI	1 Elapse	Avg						.0059	.0000	.0000	.0000
		Max						.0059	.0000	.0000	.0000
	Count	Avg	9	9	0	4	0	22	16	0	0
		Max	9	9	0	4	0	22	16	0	0
File	#Tasks	***** FC Calls *****							***** I/O Waits *****		Excl
		Get	Put	Browse	Add	Delete	Total	File	RLS	CFDT	
CUSTOME8	1 Elapse	Avg	.0000	.0000	.0000	.0187	.0000	.0187	.0015	.0000	.0000
		Max	.0000	.0000	.0000	.0187	.0000	.0187	.0015	.0000	.0000
	Count	Avg	0	0	0	3	0	3	5	0	0
		Max	0	0	0	3	0	3	5	0	0
PARTS8	1 Elapse	Avg	.0003	.0845	.0000	.0101	.0000	.0950	.0043	.0000	.0000
		Max	.0003	.0845	.0000	.0101	.0000	.0950	.0043	.0000	.0000
	Count	Avg	9	9	0	1	0	19	11	0	0
		Max	9	9	0	1	0	19	11	0	0

Figure 42 CICS PA Transaction File usage summary report

The CICS PA Transaction File usage summary report is generated from activating CICS Resource Class Monitoring as shown in Figure 5 on page 6 with the **RESRCE** parameter.

This report shows the types of accesses for all the files up to a maximum that is specified in the File resource limit as defined in the MCT, set by CEMT, or set by CEMN. It shows that nine Reads, nine Rewrites, and one Write were issued against PARTS8 and that the other three Writes were issued against CUSTOME8. This file was a recoverable file so nine Reads for update require nine records written to the log and forced out before the Rewrite and the four Writes require four forces and another for the Syncpoint (14 JCWAITS).

Many more reports can be produced by CICS PA from the CICS SMF 110 monitoring records. The previous samples of reports are only an example to demonstrate how useful CICS PA is for performance analysis.

Also, CICS PA can combine the IBM DB2® SMF 101 with the SMF 110 records to produce the following information:

- ▶ A CICS/DB2 view of the applications
- ▶ SMF 111 CICS Transaction Gateway statistics report
- ▶ IBM WebSphere® MQ reports by using the SMF 116 records
- ▶ SMF 88 logger record reports
- ▶ SMF 112 Omegamon XE for CICS record reports

## SMF 88 logger data

The SMF 88 records can be post-processed by using CICS PA. Figure 43 on page 31 shows an example report for the FOR.

----- IXGWITES -----				----- DELETIONS -----				
			Bytes Writn to	Count With	Count Without	Bytes After	Bytes Int Stor	
	Total	Average	Interim	DASD	DASD	Offload	w/o DASD	
	Count	Bytes	Storage	Write	Write	w. DASD	Write	
	-----	-----	-----	-----	-----	-----	-----	
Total	35397	342851K	9686	352029K	0	35019	0	341080K
Rate (/Sec)	117	1142838		1173429	0	116	0	1136935
Minimum	35397	342851K		352029K	0	35019	0	341080K
Maximum	35397	342851K		352029K	0	35019	0	341080K

Figure 43 CICS PA report for SMF 88

Figure 43 shows an extract of an SMF 88 logger report that was produced by CICS PA. For more information about how to produce and use the data in this report, see this document:

<http://publibfp.dhe.ibm.com/epubs/pdf/cpaabc52.pdf>

Looking at these records during benchmark comparisons is another form of verification to ensure that your comparisons have similar characteristics. This logger report ties in with the CICS log stream statistics and the RMF III report that is shown in Figure 28 on page 23.

## Hardware Instrumentation (HIS)

Hardware instrumentation (HIS) is available on all machines from z10 and later. HIS has a function that is known as *CPU Measurement Facility* (CMF), which allows for the collection of CPU samples and CPU counters. Samples are written to zFS, and counters are written to SMF as 113 records and to zFS.

When CMF is activated, samples are taken, by default, at a rate of 800000 per minute and the state of the instruction executing on the CP at that time is recorded. At the end of the sampling period, HIS also produces a module map for all address spaces in the LPAR. CICS also produces a detailed map for all modules/CSECTs in its address space by running the CLDM transaction. The samples and the maps can be merged to attribute CPU usage by module.

Counters can be collected independently of samples. Counters contain hardware events, such as the number of instructions executed, number of cycles used, and cache usage. The records that are written to zFS are human-readable.

Sample collection and counter collection are functions that are built into the Processor Unit (PU). These functions have minimal effect on the system when it is activated. For more information about how to set up and use HIS, see *Setting Up and Using the IBM System z CPU Measurement Facility with z/OS*, REDP-4727:

<http://www.redbooks.ibm.com/redpapers/pdfs/redp4727.pdf>

For this benchmark, only the counters were activated for the 5-minute interval.

HIS provides the following four counter sets for each CP:

- Basic Counter set:
  - CPU speed (cycles/microseconds)
  - 0: CYCLE COUNT
  - 1: INSTRUCTION COUNT
  - 2: L1 I-CACHE DIRECTORY-WRITE COUNT

- 3: L1 I-CACHE PENALTY CYCLE COUNT
- 4: L1 D-CACHE DIRECTORY-WRITE COUNT
- 5: L1 D-CACHE PENALTY CYCLE COUNT
- ▶ Problem State Counter set  
Same set as the Basic Counter set, but only for code running in a problem state
- ▶ Crypto-activity Counter set  
Central Processor Assist for Cryptographic Functions (CPACF) data
- ▶ Extended Counter set  
For a complete description of the Extended Counter set, see *The CPU-Measurement Facility Extended Counters Definition for System z10 and zEnterprise 196*, SA23-2261-02, at this website:  
<http://www-01.ibm.com/support/docview.wss?uid=sg20ab3047d13672ab6852574ce0044b445>

For our demonstration, Table 2 shows some of the data that was extracted from the HIS counters during the 5-minute measurement interval. IBM internal post-processing tools were used to extract the information from the SMF 113 records. Clients might choose to write their own post-processing tool or use tools that are already available in the marketplace.

SMF 113 records can also be used as input to the IBM capacity planning tool, IBM Processor Capacity Reference (zPCR), which can predict capacity based on the IBM Large Systems Performance Reference (LSPR). For more details about workload capacity performance, see this website:

<https://www-304.ibm.com/servers/resourceLink/lib03060.nsf/pages/lspwork?OpenDocument>

Table 2 Data extracted from the HIS counters

Type	Data
CPU speed	5504 cycles per microsecond
Number of cycles used	1415883143928
Number of instructions	369001248872
L1 instruction cache miss count	17766669041
L1 data cache miss count	11156580125
Relative nest intensity	0.35
Sourced from L2	66.53%
Sourced from L3	32.18%
Sourced from local L4	0.76%
Sourced from remote L4	0.10%
Sourced from local real	0.03%
Sourced from remote real	0.21%
Average cycles per instruction (CPI)	3.83



MIPs per CP	1437
L1 cache miss rate	7.83%
LPAR busy	28.6%

Table 2 on page 32 shows data that was extracted from the HIS counters that were written to SMF. All of this data is relative to the LPAR, not to any single address space.

The rows in *red* represents data that was extracted from the basic set of counters. The rows in *blue* represent data that was calculated from the extended set of counters. And, the rows in *green* were calculated by using the Basic Counter set.

The following example shows these calculations:

$\text{CPI} = \text{Number of cycles used} / \text{Number of instructions}$

$\text{MIP rate per CP} = 5504000000 / \text{CPI}$

$\text{LI Cache miss rate} = (\text{L1 instruction cache misses} + \text{L1 data cache misses}) / \text{Number of instructions}$

$\text{LPAR busy} = (\text{Number of cycles used} / \text{interval in seconds}) / (5504000000 * 3 \text{ CPs})$

When you compare benchmarks, this data can be useful in determining, for example, whether the two runs have similar characteristics in terms of any other work on this LPAR or other LPARs that affect the performance of one of the runs more than the other. If other work is interfering, it might disrupt the CP caches and so the MIP rate is reduced and the CPIs increase. A good test is whether the CPI rates that are reported by HIS are similar for both runs.

## Summary

This paper attempted to show effective ways of collecting performance data for CICS benchmarks or for capacity planning. It shows how the data that is collected from various places ties together.

The following points are key:

- ▶ The CPU/transaction can vary based on outside influences that are not always obvious.
- ▶ The hardware needs to be dedicated to the benchmark for consistent results.
- ▶ Usage of the z/OS WLM report classes is important to monitor CPU and transaction rates effectively.
- ▶ You need to measure multiple times to ensure consistency and look for trends.
- ▶ Usage of the RMF or CICS statistics for CPU comparisons rather than transaction CPU times from CICS Performance Class Monitoring is important because CICS Monitoring does not account for all of the CPU time.
- ▶ Usage of the Hardware Instrumentation (CPU Measurement Facility) is important to validate the data consistency.

## Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

**John Burgess** is a Senior Software Engineer working in the IBM Hursley CICS performance team. He has over 20 years experience with CICS and specializes in the performance of large systems.

**Manuela Mandelli** is a Senior Product Specialist in IBM Italy, based in Vimercate, Milan. She has 27 years of experience in supporting CICS. She studied in Liceo Scientifico in Milano. Her areas of expertise include CICS Transaction Server, CICS Transaction Gateway, and IBM Session Manager.

Thanks to the following people for their contributions to this project:

Rufus P. Credle Jr.  
International Technical Support Organization, Raleigh Center

Chris Baker, Catherine Moxey, Arndt Eade, Ian Burnett, Graham Rawson, George Burgess  
IBM Hursley

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:  
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:  
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:  
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:  
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:  
<http://www.redbooks.ibm.com/rss.html>

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright International Business Machines Corporation 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This document REDP-5170-00 was created or updated on February 17, 2015.

Send us your comments in one of the following ways:


- ▶ Use the online **Contact us** review Redbooks form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an email to:  
[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.



## Trademarks

IBM, the IBM logo, and [ibm.com](http://ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

CICS®	MVS™	RMF™
CICSplex®	Redbooks®	System z®
DB2®	Redpaper™	WebSphere®
IBM®	Redbooks (logo)  ®	z/OS®
IMS™	Resource Measurement Facility™	z10™

Other company, product, or service names may be trademarks or service marks of others.