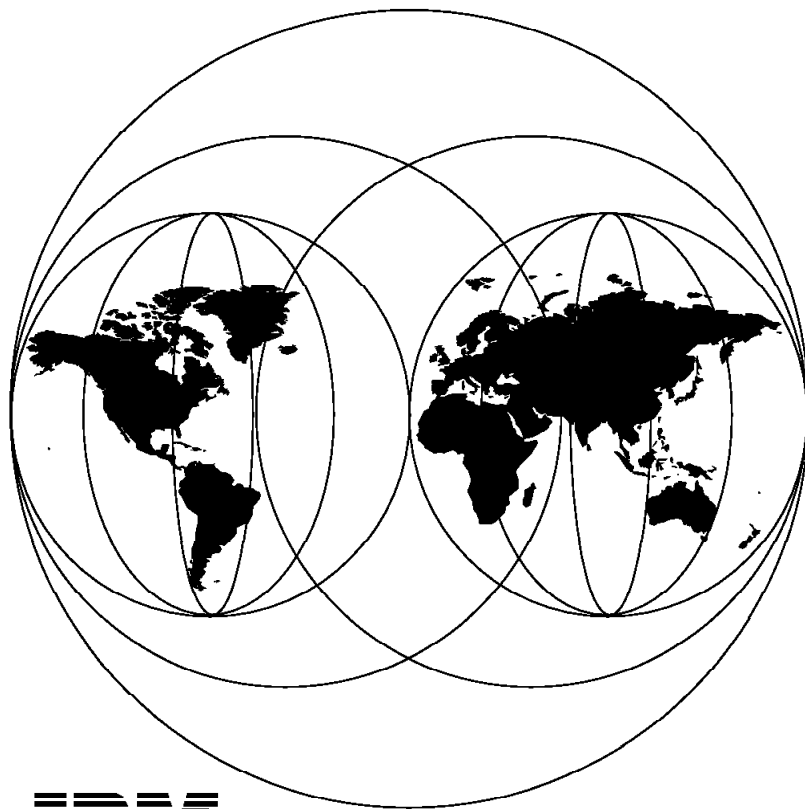


International Technical Support Organization

SG24-4352-00

**System/390 MVS/ESA Version 5
Workload Manager Performance Studies**

September 1995



IBM

**International Technical Support Organization
Poughkeepsie Center**



International Technical Support Organization

SG24-4352-00

**System/390 MVS/ESA Version 5
Workload Manager Performance Studies**

September 1995

Take Note!

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xiii.

First Edition (September 1995)

This edition applies to Version 5 Release 1 of MVS/ESA System Product (5655-068 or 5655-069).

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1995. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Abstract

This document is unique in its detailed coverage of the MVS workload manager, the system resource manager (SRM), and the new functions of Resource Measurement Facility (RMF) Version 5. It provides performance analysis of the new MVS workload manager function and the modified SRM component in MVS/ESA Version 5 Release 1.0. Together, they dynamically manage the performance of work in systems that are part of a sysplex.

The MVS workload manager also provides services for subsystems such as CICS/VS and IMS/VS to identify and control their transactions. Services are also provided to track the response times of the various workloads, and provide this information both to third party monitor programs and to IBM-supplied monitor programs, such as Resource Measurement Facility (RMF).

This document was written for anyone involved in the task of measuring and improving the performance of MVS systems or anyone who needs an understanding of this task. Some knowledge of MVS and RMF and how MVS systems work is assumed.

(136 pages)

Contents

Abstract	iii
Special Notices	xiii
Preface	xv
How This Document is Organized	xv
Related Publications	xv
International Technical Support Organization Publications	xvi
ITSO Redbooks on the World Wide Web (WWW)	xvi
Acknowledgments	xvii
Chapter 1. Introduction to Workload Management	1
1.1 Workload Management	1
1.1.1 MVS/ESA 5.1.0	2
1.1.2 System Resource Manager	2
1.1.3 Cross-System Coupling Facility Services	2
1.1.4 The ISPF Application	3
1.1.5 RMF Monitoring and Management	3
1.1.6 CICSplex/System Manager (CP/SM)	3
1.1.7 IMS, TSO, and Batch	4
1.2 Workload Manager Terminology	4
1.2.1 Service Class	4
1.2.2 Goals	4
1.2.3 Periods	4
1.2.4 Duration	5
1.2.5 Importance	5
1.2.6 Response Time	5
1.2.7 Velocity	5
1.2.8 Discretionary	6
1.2.9 Resource Groups	6
1.2.10 Service Policy	6
1.2.11 Performance Index	6
1.2.12 Subsystem Work Managers	6
1.3 Why Migrate to Goal Mode?	7
Chapter 2. System Resource Manager MVS/ESA Version 5	9
2.1 SRM Structure and Algorithms	10
2.1.1 Performance Index Calculation	11
2.1.2 Dispatching Priorities	14
2.1.3 Target Multiprogramming Levels	14
2.1.4 Swap Protect Time	15
2.1.5 Storage Targets	15
2.1.6 Expanded Storage Policy	16
2.1.7 Capping	16
2.1.8 Server Topology	17
2.1.9 Donors and Receivers Determination	17
2.2 WLM Service Routines	18
2.2.1 Work Manager Services	19
2.2.2 Execution Delay Monitoring Services	20
2.2.3 Performance Block States	24
2.2.4 Server Topology Assessment	27

2.2.5 WLM Services / IMS Example	29
2.2.6 WLM Services CICS / CPSM / DL1 Example	30
2.2.7 Workload Reporting Services	34
2.3 Policy Adjustment	34
2.3.1 Donors and Receivers Concept	34
2.3.2 Policy Adjustment Loop	36
2.4 Resource Adjustment Routine	50
2.4.1 Concepts in Resource Adjustment Routine	50
2.4.2 Address Space Monitoring	51
2.4.3 Resource Adjustment Actions	52
2.4.4 System Over-Utilized	52
2.4.5 System Unmanaged CPU Paging Cost High	53
2.4.6 System Auxiliary Paging High	53
2.4.7 OK1 Action	55
2.4.8 System is Under-Utilized	55
2.4.9 Phase Change	56
2.4.10 Working Set Management	56
Chapter 3. Workload Manager Performance Monitoring	59
3.1 RMF Reports in Compatibility Mode	60
3.1.1 Workload Activity Report-Compat Mode	60
3.2 RMF Reports in Goal Mode	61
3.2.1 RMF Monitor III Reports in Goal Mode	65
3.3 SMF Record Type 99	67
3.3.1 Identifying Work in SMF type 99 Records	68
3.3.2 Identifying Server Service Classes	68
3.3.3 Identifying Internal Service Classes	68
3.3.4 Interpreting SMF Record Type 99 Traces	69
3.3.5 Action Trace Example	70
3.3.6 Policy Adjustment/Dispatching Priority Example	71
3.3.7 Resource Adjustment/Working Set Management Example	73
3.3.8 Resource Adjustment/System Under-Utilized Example	77
3.3.9 Policy Adjustment/Cap Slices Example	78
Chapter 4. Workload Manager Test Cases	81
4.1 TSO Workload	81
4.1.1 TSO goals	82
4.1.2 Results	83
4.1.3 Observations	85
4.2 Another TSO Workload	85
4.2.1 Results	86
4.2.2 Observations	86
4.3 Storage Stress Environment	87
4.4 Mixed Workload with CICS More Important	91
4.5 Capping Test Cases	96
4.5.1 Test Case with a Maximum	96
Chapter 5. Migration Hints and Tips	99
5.1 Software Positioning	99
5.2 How to Obtain Goals	99
5.2.1 TSO Workload	99
5.2.2 Batch Workload	100
5.2.3 CICS and IMS Workload	100
5.2.4 Started Tasks and Other System Address Spaces	101
5.2.5 APPC	103

5.2.6 Capping	103
5.3 Running in Goal Mode	103
5.4 General Recommendations and Observations	103
5.5 Summary	104
Appendix A. IEAIPS Member used in the WLM Test Cases	107
Appendix B. IEAICS Member used in the WLM Test Cases	111
Appendix C. IEAOPT Member used in the WLM Test Cases	119
Appendix D. Service Definition — PERFPOL	121
D.1 Overview	121
D.1.1 Service Coefficients	121
D.2 Classification Rules	121
D.3 Workload/Service Class Descriptions	126
D.4 Report Classes	126
D.5 Service Definition Service Class Goals	130
Index	133

Figures

1.	Workload Management Environment	2
2.	ISPF Administration Application	3
3.	Workload Definitions in Terms of Business Goals	7
4.	New WLM and SRM Structure	9
5.	Response Time Distribution	13
6.	BTE and EXEC Phase	23
7.	BTE and EXEC Phases	24
8.	Server Topology Example	29
9.	Timetable of Events	33
10.	Some RMF Data	34
11.	Loop Summary	36
12.	Period Wide Paging Plot	41
13.	Address Space Paging Plot	42
14.	Period MPL Delay Plot	44
15.	Period Ready User Average Plot	45
16.	Period Swap Delay Plot	47
17.	Paging Responsiveness	54
18.	Workload Activity Report in Compatibility Mode	60
19.	ASRM Report in Goal Mode	61
20.	Workload Activity Report in Goal Mode	63
21.	Sysplex Performance Summary Report	65
22.	Response time Distribution Report	66
23.	Work Manager Delays Report	67
24.	Subtype 1 Trace Data Output	70
25.	Subtype 2 Trace data output	70
26.	DP Period Summary First Example	72
27.	DP Trace data output	72
28.	DP Period Summary Second Example	73
29.	WSM Period Summary	74
30.	WSM First Trace Output	74
31.	WSM Second Trace Output	75
32.	WSM Third Output Trace	76
33.	System Under-utilized First Period Summary	77
34.	System Under-Utilized Trace Output	77
35.	System Under-Utilized Second Period Summary	77
36.	Resource Group Data	79
37.	Capping Action Trace	80
38.	TSO Workload in Compatibility Mode	82
39.	TSO Workload in Goal Mode	83
40.	Batch Storage Stress	88
41.	Batch Storage Stress	89
42.	RMF Paging Activity Report for Storage Stress Environment	90
43.	CICS More Important Test Case	93
44.	Performance Index for CICS Important Test Case	94
45.	SMF 99 Trace for CICS Important Workload	95
46.	Consumed Service by Workload and Time	97
47.	TSO Performance Index by Time	98

Tables

1.	Central Storage Plots First Example	75
2.	Processor Storage Plots First Example	75
3.	Central Storage Plots Second Example	75
4.	Processor Storage Plots Second Example	76
5.	Central Storage Plots Third Example	76
6.	Processor Storage Plots Third Example	76
7.	Goals for TSO	83
8.	TSO Test Results	84
9.	Goal for Second TSO Measurement	86
10.	Goals for Storage Stress Environment	87
11.	Paging Statistics for Batch Storage Stress Jobs	91
12.	Goals for Mixed Workload with CICS More Important	92
13.	CICS Classification Rules	121
14.	IMS Classification Rules	123
15.	JES Classification Rules	123
16.	STC Classification Rules	124
17.	TSO Classification Rules	125
18.	Service Definition Goals Sorted by Service Class	130
19.	Service Definition Goals Sorted by Workload	130

Special Notices

This publication is intended to help systems programmers and performance personnel who work with performance monitoring reports to understand the performance of an MVS system and to plan for the appropriate tuning activities. The information in this publication is not intended as the specification of any programming interfaces that are provided by MVS/ESA Version 5 Release 1.0. See the PUBLICATIONS section of the IBM Programming Announcement for MVS/ESA Version 5 Release 1.0 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

ACF/VTAM	CICS
CICS/ESA	DATABASE 2
DB2	ES/9000
ESCON	Hiperspace
IBM	IMS/ESA
MVS/ESA	OPC
RACF	Resource Measurement Facility
RMF	Sysplex Timer
VTAM	

The following terms are trademarks of other companies:

Windows is a trademark of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other trademarks are trademarks of their respective companies.

Preface

This document is intended for personnel who work with performance tuning of MVS systems. It provides general information about the MVS Workload Manager (WLM), a new component of the MVS/ESA Version 5 Release 1.0 BCP; the System Resource Manager (SRM), a component of MVS which is significantly different than earlier MVS Releases; and the Resource Measurement Facility (RMF) Version 5.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "Introduction to Workload Management" contains an overview of the new concept of MVS workload management, the specification of *goals* for the system workload.
- Chapter 2, "System Resource Manager MVS/ESA Version 5" describes the systems resource manager (SRM) and the algorithms it uses.
- Chapter 3, "Workload Manager Performance Monitoring" describes monitoring the system to track the attainment of the workload's goals. The various actions that the workload manager takes to ensure the workload meets its goals are traced in SMF 99 trace records.
- Chapter 4, "Workload Manager Test Cases" discusses some test cases designed to investigate the behaviour of the WLM with different workloads.
- Chapter 5, "Migration Hints and Tips" gives some hints and tips on how to migrate to goal mode.
- Appendix A, "IEAIPS Member used in the WLM Test Cases" contains the IPS parameters used in the compatibility mode runs.
- Appendix B, "IEAICS Member used in the WLM Test Cases" contains the ICS parameters used in the compatibility mode runs.
- Appendix C, "IEAOPT Member used in the WLM Test Cases" contains the OPT parameters used in the compatibility mode runs.
- Appendix D, "Service Definition — PERFPOL" contains the policy definitions that were used in the goal mode runs.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *CICSplex System Manager Concepts and Planning*, GC33-0786
- *MVS/ESA Planning: Workload Management*, GC28-1493
- *MVS/ESA Programming: Workload Management Services*, GC28-1494
- *MVS/ESA System Management Facilities*, GC28-1457
- *MVS/ESA SP Version 4 Technical Presentation Guide*, GG24-3594-01
- *MVS/ESA Initialization and Tuning Reference*, SC28-1452
- *MVS/ESA Resource Measurement Facility Version 5 User's Guide*, GC33-6483

The following publications are available to IBM-licensed customers only.

- *MVS/ESA Resource Measurement Facility Version 5 Getting Started on Performance Management*, LY33-9176
- *MVS/ESA Analyzing Resource Measurement Facility Version 5 Reports*, LY33-9178

International Technical Support Organization Publications

The following are publications you may find useful.

- *MVS/ESA 5.1.0 Announcement*, GG24-4137
- *MVS/ESA 5.1.0 Implementation Guide*, GG24-4354
- *Parallel Sysplex Performance*, GG24-4356
- *Parallel Sysplex Operations*, GG24-4370
- *Parallel Sysplex Implementation*, GG24-4386

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

Bibliography of International Technical Support Organization Technical Bulletins, GG24-3070.

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOPUB LISTALLX. This package is updated monthly.

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called GBOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser (such as WebExplorer from the OS/2 3.0 Warp BonusPak) to the following:

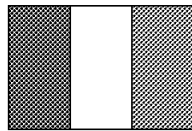
<http://www.redbooks.ibm.com/redbooks>

IBM employees may access LIST3820s of redbooks as well. Point your web browser to the IBM Redbooks home page:

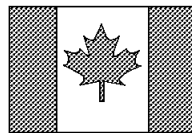
Acknowledgments

This publication is the result of a residency conducted at the International Technical Support Organization, Poughkeepsie Center.

The advisors for this project were:



Christian Matthys
International Technical Support Organization,
Poughkeepsie

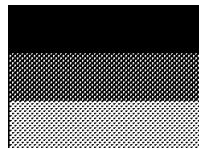


G. Tom Russell
International Technical Support Organization,
Poughkeepsie

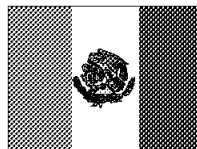
The authors of this document are:



Alvaro Salla
IBM Brazil



Dr. Roland Wolf
IBM Germany



Gustavo Gonzalez
IBM Mexico

Many thanks to the following people. Not only did they perform the tests that are documented here, but they also provided invaluable guidance and advice, including a significant amount of editing support of the manuscript.

John Arwe	IBM S/390 Division Poughkeepsie
Ed Berkel	IBM S/390 Division Poughkeepsie
Cathy Eilert	IBM S/390 Division Poughkeepsie
Peter Enrico	IBM S/390 Division Poughkeepsie
Steve Grabarits	IBM S/390 Division Poughkeepsie
Mel Short	IBM S/390 Division Poughkeepsie
Peter Yocom	IBM S/390 Division Poughkeepsie

Chapter 1. Introduction to Workload Management

With MVS/ESA Version 5, a new concept is introduced for workload and resource management. The system manages the access to resources according to the goals set by the installation.

1.1 Workload Management

In the past the tuning of an MVS system was a very difficult task; only a few people had the necessary skill to do it efficiently. Tuning could be compared to a magician using magic spells to help the poor users who were suffering bad response time. A lot of parameters with strange names had to be adjusted to achieve the goal of good response time for one application without harming the response of other applications.

MVS workload management as implemented in MVS/ESA Version 5 allows you to think about the goals (good response time) not the strange sounding parameters. For example, a response time goal for TSO could be stated as "TSO trivial transactions should have an average response time of .25 seconds." This is all you need (or can) do. It is the task of the MVS workload manager, not the system programmer, to set the internal parameters in a way to achieve this goal. Workload management controls the distribution of resources in accordance with the goal specifications (response time) and at the same time aims for the best utilization of those resources (throughput).

The new workload management introduces a new concept in workload control: the workload is defined in terms of *business service level goals*. Two MVS workload manager modes are provided:

- Compatibility mode, where the system is managed according to parameters in the IPS and ICS
- Goal mode, where the WLM manages the system according to the goals specified in the *policy* specified by the installation, and the IPS and ICS are ignored

Workload management activities are functionally distributed among the following major components and products as shown in Figure 1 on page 2:

- Workload Manager (WLM) BCP MVS/ESA Version 5 Release 1
- WLM Services BCP MVS/ESA Version 5 Release 1
- SRM BCP MVS/ESA Version 5 Release 1
- VTAM Version 4 Release 2.0
- CICS/ESA Version 4 Release 1.0
- CP/SM Version 1 Release 1.0
- IMS/ESA Version 5 Release 1.0
- RMF Version 5 Release 1.0

With this release, MVS WLM is aware of transactions running in the CICS and IMS address spaces. New versions of CICS and IMS are required to provide this information to WLM. Only CICS/ESA Version 4 Release 1.0 with CICSplex SM, and IMS/ESA Version 5 Release 1.0 participate today in the workload

management concept as defined in MVS/ESA Version 5 Release 1.0. Levels of CICS or IMS earlier than the above will not tell WLM about their transactions. Therefore WLM will manage these subsystems based on address space goals.

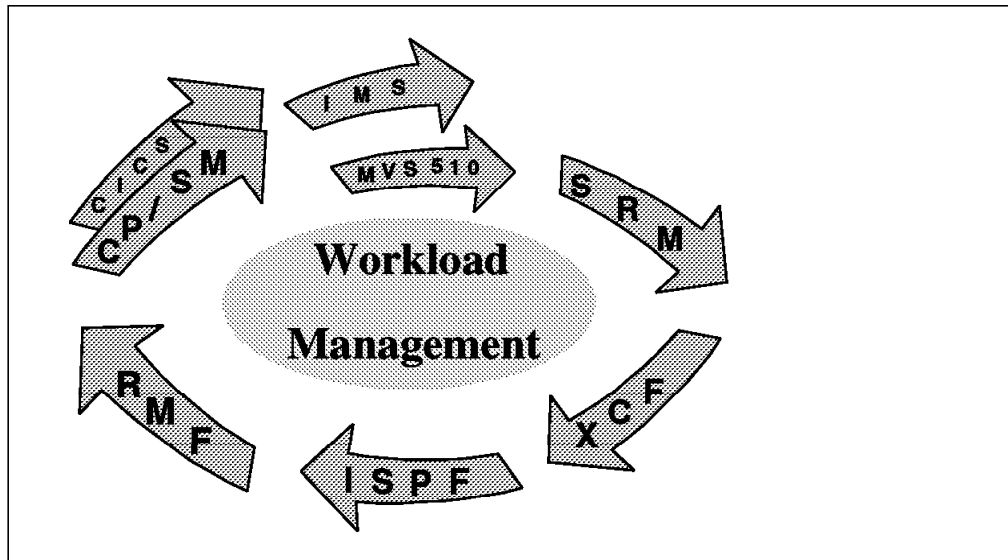


Figure 1. Workload Management Environment

1.1.1 MVS/ESA 5.1.0

The MVS Workload Manager is a new system component included in the MVS/ESA Version 5 Release 1.0 Base Control program (BCP). The workload manager component is automatically activated during the initial program load (IPL). The workload manager functions exercise control over CPU and storage resources.

1.1.2 System Resource Manager

The existing System Resource Manager (SRM) component contains new algorithms for the MVS Workload Manager. The SRM also allows dynamic switching between the traditional SRM IPS/ICS mode of operation (*compatibility mode*) and the goal-oriented mode (*goal mode*).

1.1.3 Cross-System Coupling Facility Services

Workload management introduces a sysplex-wide view of performance administration, monitoring, and management of workloads running under MVS. The MVS workload manager runs in a sysplex, with a workload manager function couple data set (WLM CDS) shared among all the systems. All the systems are informed of any change in the active service definition. The workload manager couple data set allows high availability: in case of a failure in the path or in the disk itself, the alternate data set is automatically activated.

The MVS workload manager uses XCF services to communicate. There is no requirement for a coupling facility hardware unit; ESCON channel-to-channel (CTC) connections can be used.

Each processor in a sysplex uses a common service definition, or *policy*. The policy is stored in the shared workload manager couple data set, and may be refreshed by the service administrator or authorized system operator. Activating

a new policy on one system in the sysplex results in all other systems activating the same new policy simultaneously.

1.1.4 The ISPF Application

ISPF application allows the service administrator:

- To enter the MVS workload manager service definition into the system
- To install the definition in a data set called the *MVS workload manager couple data set* (WLM CDS)
- To activate a definition for the sysplex

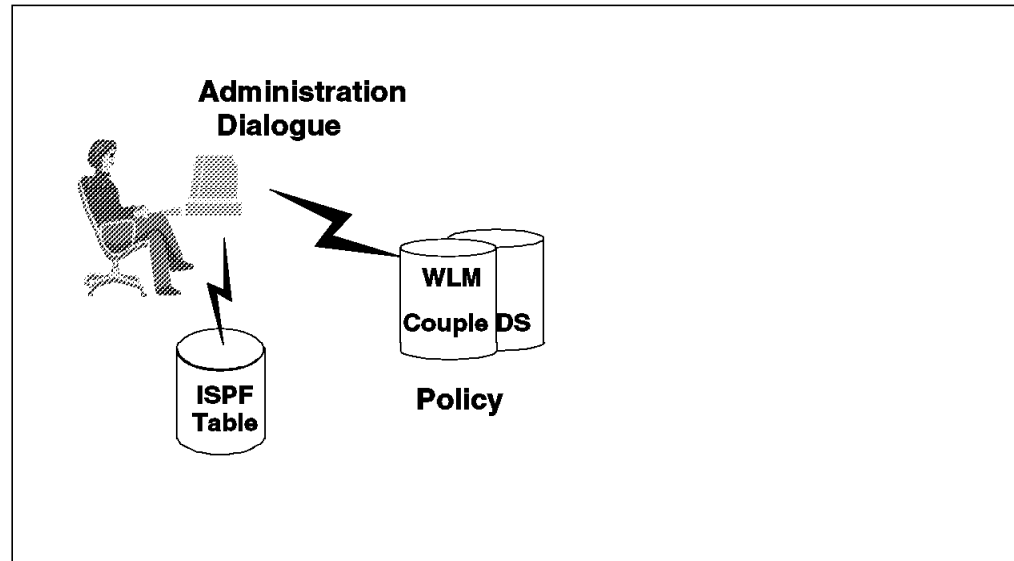


Figure 2. ISPF Administration Application

1.1.5 RMF Monitoring and Management

Each system in goal mode in the sysplex routinely exchanges performance monitoring data with the others in the sysplex using XCF. This information includes goal data, such as response time and delays. Information as to how well each system is doing in processing towards a goal is recorded in SMF records on each system. This data, after being processed by the reporter, such as RMF, can then be used to report the resources used, enabling a sysplex-wide view of performance data to be obtained from any system in the sysplex. RMF Monitor I and Monitor III provide sysplex-wide view reports from any MVS.

Note: The MVS workload manager does not by itself provide a sysplex-wide view for goal and delay data. The reporter has to compile the data before providing the sysplex-wide view reports.

1.1.6 CICSplex/System Manager (CP/SM)

The first subsystem able to fully use the workload management functions is CICS/ESA Version 4 Release 1.0. The workload management functions provided in that product are also enhanced by a new product, CICSplex System Manager (CICSplex SM). CICS/ESA Version 4 Release 1.0 uses WLM services to *classify* each arriving transaction into a *service class*. With this and other new WLM services, the BCP knows about CICS transactions, their response time, and which regions are processing which transactions. CP/SM retrieves the goal for

the transaction's service class from the MVS workload manager. It measures the actual performance of each transaction in each CICS address space; thus, it is able to adjust the transaction routing to distribute work to CICS address spaces in such a way that the goal response time may be achieved.

In other words, in a CICS environment with CP/SM active, it is possible to route arriving work to any one of a number of CICS regions which are capable of processing a particular CICS transaction. CP/SM routes transactions to the region best capable of processing the work, based on the goals as defined to the MVS workload manager for all the transactions.

1.1.7 IMS, TSO, and Batch

Other work, such as IMS transactions, TSO transactions, or batch jobs, can be classified into a service class based on the service definition in the policy. The MVS workload manager allocates CPU and storage resources to the address spaces as needed to achieve the goals for these workloads, and RMF reports the results attained based on these goals. Unlike CICS, there is no dynamic routing or *workload distribution* function in MVS/ESA Version 5 Release 1.0 for these transactions.

1.2 Workload Manager Terminology

There are no performance groups or domains in MVS when the system is running in *goal mode*. They are replaced by *service classes*.

1.2.1 Service Class

A service class is similar in concept to a performance group. A service class represents a grouping of work with similar resource requirements and performance requirements, which can be address spaces or CICS/IMS transactions. Each service class can have one or more *periods*. Each service class period has a *goal*.

1.2.2 Goals

There are different goal types, which will be discussed later:

- Average response time
- Response time with percentile
- Execution velocity
- Discretionary

1.2.3 Periods

The workload manager provides *performance periods* that you can use to specify a series of varying goals and importance. Up to eight performance periods for each service class can be defined. Each performance period (except the last period) has a *duration* in service unit terms.

1.2.4 Duration

The duration specifies the length of the period in service units. If the work does not complete when the number of service units has been used, the work moves to the next performance period. Do not specify a duration for the last period. Multiple performance periods are not available for work in the IMS and CICS subsystem work environments.

Each performance period has an *importance* that indicates how vital it is to meet the service goal.

1.2.5 Importance

The *importance* is the relative importance of the service class goal. Importance is used when work is not meeting its goal. It helps determine which service class should donate resources to a higher importance service class that is not achieving its goal. Importance levels are represented internally by numbers from 1 to 5, where 1 is the most important.

1.2.6 Response Time

Response time is the expected amount of time required to complete the work submitted under the service class, including queue time. Specify either an *average response time* or a *response time with a percentile*.

- For a response time with percentile goal, you specify the percentage of transactions that should complete during the time.
- For an average response time, you specify the total average response time for all the transactions in the service class period.

1.2.7 Velocity

Velocity is the measure of how fast work should run when ready, without being delayed for processor or storage access. Velocity is expressed as a percentage from 1 to 99. The formula for velocity is:

$$velocity = \frac{CPU\ using\ samples}{(CPU\ using + CPU\ delay + STOR\ delay + MPL\ delay)\ samples}$$

In other words, the velocity goal defines the acceptable amount of delay for work when work is ready to run. Velocity goals should be used for long-running jobs and for address spaces; they are the most appropriate goal for any started tasks that require a goal.

Velocity is also appropriate for the *server* address spaces, that is, the address spaces that work on behalf of a transaction manager or a resource manager, such as CICS AOR or IMS control region. The velocity goal assigned to a server address space applies only during address space start up; after that, WLM manages resources according to the response time goals defined for the transactions in the servers.

1.2.8 Discretionary

A discretionary goal is a WLM-defined goal. This goal should be used for work without a specific service goal. A discretionary goal is treated as if the work had no *importance*.

1.2.9 Resource Groups

There is also a way to either limit or guarantee the CPU service for one or more service classes. To do this, a *resource group* can be defined. For a resource group you can specify the minimum and/or maximum CPU service (CPU plus SRB time adjusted by CPU speed, without the service coefficients) per second that the workload is allowed to consume. A *minimum* service would be specified to guarantee service to a workload. A *maximum* service specification would be used to limit the resources consumed by a workload.

1.2.10 Service Policy

The service classes, periods, goals and resource groups collectively make up a *service policy*. You can have different policies for different time periods; for example you could have one policy for the day and another for the night or one for week days and another for week ends. This is to reflect the different goals you have for your workloads at different times.

As many policies as desired can be defined. However, only one policy can be active in the sysplex at one time. If a new policy is activated, it will become active on all the systems in the sysplex.

1.2.11 Performance Index

When we define goals for a workload we also need some feedback whether the goals are being met or not. RMF Version 5 reports both the actual response time and velocity attainment as well as the goals.

The ratio of actual versus goal is called the *performance index*. It is also reported by RMF. See 2.1.1, "Performance Index Calculation" on page 11 for details.

1.2.12 Subsystem Work Managers

The subsystems CICS and IMS use new workload manager services that allow them to report response times of CICS or IMS transactions to the MVS workload manager. They also report delays that the transactions suffer. This allows us to look inside IMS or CICS, and get feedback on transaction execution in the RMF report.

The view is not at the CICS or IMS regions but at the transactions. This is a new concept. You no longer have to be (or can be) too involved with SRM parameters for CICS regions. The workload manager tunes the *server* address spaces to achieve the goals for the transactions.

1.3 Why Migrate to Goal Mode?

With MVS/ESA Version 5 Release 1 it is still possible to run the system with parameters defined in IEAIPSxx, IEAICSxx, and IEAOPTxx. This is called *compatibility mode*. Regardless of how much effort you have spent defining the various parameters in the IPS and ICS to provide a good balance of workload between the response expectations of your users and good resource utilizations, the parameters are effectively unchangeable between tuning projects. The workload manager in goal mode reacts quickly to changes in the workload and will attempt to provide the same tuning balance on an instantaneous basis.

The Workload Manager introduces a new concept in workload control in that the workload is defined in terms of *business service level goals* as shown in Figure 3.

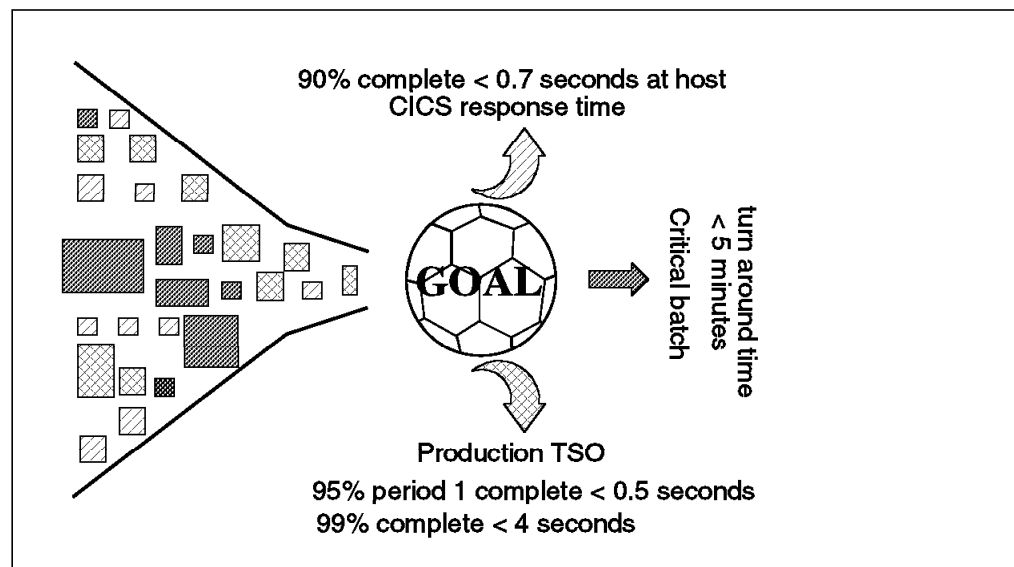


Figure 3. Workload Definitions in Terms of Business Goals

The new workload management concept addresses the need for distributing and balancing workloads which compete for processor resources, such as CPU and storage. Additionally, the concept is to define goals for work, using business terms, and to let the system decide how much work should run and how much resources to supply to meet the goals.

To ease implementation and provide fall-back capability, the MVS workload manager allows you to dynamically switch the system back to *compatibility mode*, which allows you to continue to run using the existing familiar system definitions.

Whenever the workload in your system changes you must tune and balance the system. When your system is running in goal mode the workload manager will do it for you.

Service classes and goals apply sysplex wide. Managing the relative performance of multiple workloads on multiple MVS systems can become very complicated. The MVS workload manager can handle resource allocation to all work arriving at an MVS image. Subsystem workload managers, such as CP/SM, can build on that to provide effective workload distribution for all systems in the sysplex.

In this book we want to encourage you to migrate your system to goal mode. The following chapters contain:

- Chapter 2, “System Resource Manager MVS/ESA Version 5” on page 9 describes the systems resource manager (SRM) and the algorithms it uses.
- Chapter 3, “Workload Manager Performance Monitoring” on page 59 describes monitoring the system to track the attainment of the workload’s goals.
- Chapter 4, “Workload Manager Test Cases” on page 81 discusses some test cases designed to investigate the behavior of the WLM with different workloads.
- Chapter 5, “Migration Hints and Tips” gives some hints and tips on how to migrate to goal mode.

Chapter 2. System Resource Manager MVS/ESA Version 5

The WLM works like a thermostat in a house. You set the required temperature and the WLM adjusts gauges, valves, furnace and so on in order to keep the temperature at the value you set. You do not normally need to know how the system works internally. However, there is one exception — when the goal is not being achieved and you start to freeze. In this case you need to know more to survive.

This is the major reason for this chapter; to explain what the system resource manager (SRM) does to match the goals passed by WLM. We hope that at completion you will be comfortable in tackling a workload activity RMF report in goal mode and even a formatted output of the SMF 99 record.

Another key point for this chapter is the assumption that the reader is conversant with the previous version of the SRM. Previous knowledge of concepts such as swapping, storage isolation, and dispatching is mandatory.

The SRM is BCP code running in the nucleus without an address space of its own, and a WLM address space is activated at IPL, as shown in Figure 4.

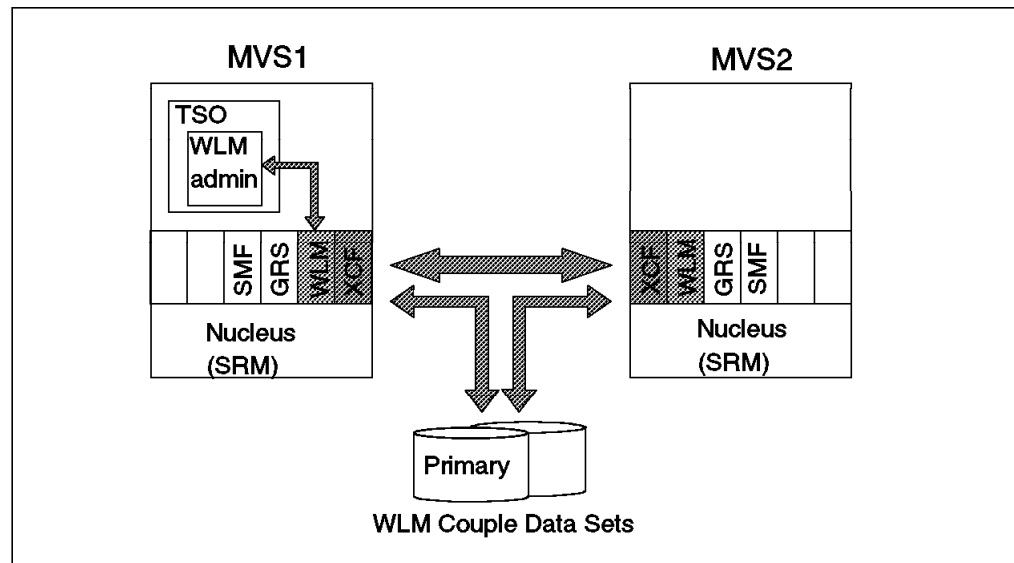


Figure 4. New WLM and SRM Structure

In MVS/ESA Version 5 Release 1, the SRM belongs to a set of functions called workload management.

The SRM can run in two modes: *compatibility* and *goal* mode. In compatibility mode, all the parameters defined in the OPT, ICS, and IPS members are valid with the exception of APGRNG; APGRNG always defaults to (0,15).

There are some structural differences in the SRM in MVS/ESA Version 5 Release 1 when compared with previous releases, but the primary difference is that in goal mode, the SRM algorithms are driven by WLM using the installation goals and not by the former IPS and ICS PARMLIB constructs.

This chapter covers the SRM in goal mode. Also, WLM services are extensively described due to their many interfaces with the SRM.

2.1 SRM Structure and Algorithms

Structurally, the SRM logic in goal mode is composed basically of two main routines:

- Policy adjustment, discussed in 2.3, “Policy Adjustment” on page 34.
- Resource adjustment, discussed more thoroughly in 2.4, “Resource Adjustment Routine” on page 50.

Disregarding I/O, there are two basic resources of a system that are required in order to execute units of work as tasks (represented by TCBs) and service requests (represented by SRBs). These resources are the processor and the processor storage. SRM manages these resources through the algorithms and by measuring the various delays that units of work are experiencing.

When this book uses the expression “manage an address space” we are referring to the process of tracking the delay states, and granting or denying access to the systems resources. There are only three states used for tracking delays:

CPU being ready to run, but not being dispatched because of a higher dispatching priority item.

STOR Being delayed for a page-in operation, either from Auxiliary or Expanded Storage.

MPL Being swapped out, either logically or physically with no MPL available for a swap in.

SWIN Being swapped out physically with an MPL available for a swap in.

Address spaces are managed to their goals. If an address space is behind (not achieving its goal), and the primary delay is storage, then the address space will (probably) be given storage.

SRM controls the use of storage through storage targets, a swap protect time target, the MPL swap mechanism and expanded storage policies.

In this chapter the algorithms are introduced to familiarize the reader with new ideas and the new words of SRM and WLM in goal mode. When possible, correlations are established with the compatibility mode concepts. A more detailed coverage of the algorithms is presented in the remainder of this chapter.

Note: I/O priority queueing and the I/O “boundness” concept in mean-time-to-wait algorithm are the only decisions related with the I/O resources that are taken by SRM in MVS/ESA Version 5 Release 1.

The following routines execute the major SRM algorithms:

- Performance index calculation
- Dispatching priority assignment
- MPL targets
- Swap protect time
- Storage targets
- Expanded storage policy
- Capping

- Server topology
- Donors and receivers determination

2.1.1 Performance Index Calculation

The Performance Index (PI) indicates how well a service class is doing in meeting its goal. Each service class period may have two types of PIs:

Sysplex PI A PI that represents its global performance across all the systems in the sysplex.

Local PI A PI that represents its performance in each local system of the sysplex. A service class may have more than one local PI if that service class has work executing on multiple systems in the sysplex.

Through services provided by SRM, performance data is periodically exchanged between sysplex systems in goal mode. The information is sent across the systems and allows SRM to construct an approximate view of the status of the sysplex through the calculation of sysplex PIs. The aggregate view enables SRM algorithms such as policy adjustment to make trade-offs within each individual system. This exchanged data includes:

- Response time information for each service class period, to allow each local SRM to calculate the sysplex PIs for each service class period.
- CPU service consumption information for each resource group, to enforce globally the capping and the protection algorithms.
- Information identifying the active service policy

The SRM uses the PI value in the following ways:

- When the PI is equal to one, this means that the service class period is exactly meeting its goal.
- When the PI is greater than one, this means that the service class period is missing its goal.
- When the PI is less than one, this means that the service class period is exceeding its goal.

2.1.1.1 Performance Index for Goals

The following formulas for a PI calculation for a service class period depends on the type of the goal:

- Average Response Time Goal:

$$\text{Performance index} = \frac{\text{Actual average response time}}{\text{Goal average response time}}$$

- Execution Velocity:

$$\text{Performance index} = \frac{\text{Goal execution velocity}}{\text{Actual execution velocity}}$$

- Response Time Percentile:

$$\text{Performance index} = \frac{\text{Percentile actual}}{\text{Percentile goal}}$$

Note: See 2.1.1.3, “PI Calculation for Response Time Percentile Goal” on page 13 for an explanation of this calculation.

- Discretionary:

In MVS/ESA Version 5 Release 1, a discretionary service class period is defined to have a PI equal to 0.81. A discretionary service class period is also called a universal donor, as its PI indicates that it is always beating its goal.

A PI can be calculated over any interval. If you have RMF Monitor I with a 30-minute interval, RMF calculates a PI for that interval. Monitor III shows a PI calculated over any selected range.

2.1.1.2 PI Calculation Example

The WLM keeps response time distribution data in *buckets* for service class periods that have a response time goal specified. The buckets are counters that keep the number of transactions ended within a certain response time range. The response times are stored in a structure that is represented as 14 buckets. These buckets exist per-service-class period. In Figure 5 on page 13, each *bucket* contains three pieces of information:

1. The number of the bucket, from 1 to 14.
2. The *percent of goal* refers to the response time intervals which start at 50%, or $\frac{1}{2}$ the goal in bucket 1. Transactions that completed in an elapsed time less than half the response time goal are captured in the first bucket.

Bucket 6 is 100% or equal to the goal and contains transactions that ended between 0.9 times and 1.00 times the response time goal.

Bucket 12 is twice the goal.

Bucket 13 contains counts from two to four times the goal, and bucket 14 is the number of transactions with a response time greater than 4 times the goal.

The maximum value for a PI in a percentile goal is 4.0 due to the way the buckets are organized. Bucket 13 corresponds always to 400% the value of the goal.

3. The third set of numbers for each bucket contains the number of transactions completed in each time interval and is referred to as the *transaction count*.

Note: Buckets in the 2 to 11 range are evenly distributed and serve to capture transactions that ended between the interval represented by 0.5 times the goal to 1.5 times the goal. In the example, bucket 2 contains the number of ended transactions with a response time between 1.0 and 1.2 seconds.

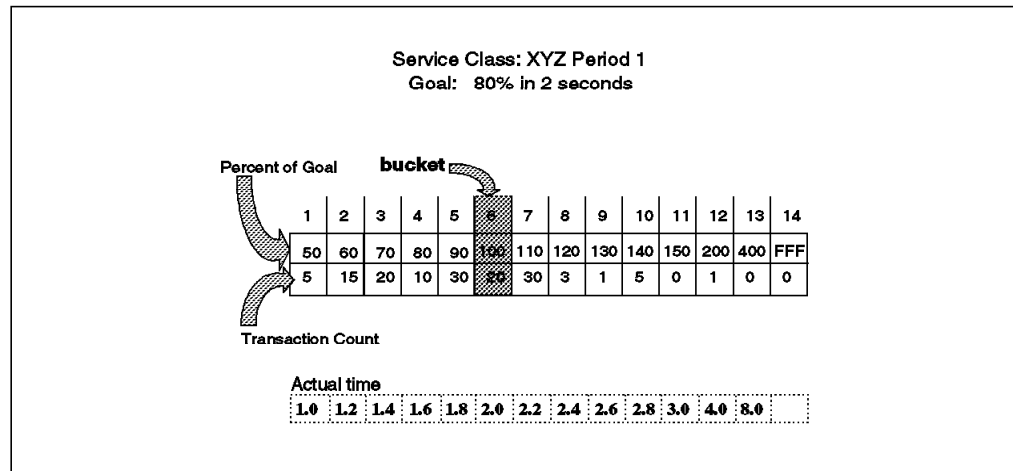


Figure 5. Response Time Distribution

When we add all the transactions that are captured in buckets 1 to 6, we know the number of transactions that achieved the goal. Using all this information, a reporter product can easily produce a response time distribution. This response time distribution is only provided in goal mode and is not reported for a report class.

2.1.1.3 PI Calculation for Response Time Percentile Goal

In the example in Figure 5, the goal for the first period is:

A response time of 2.0 seconds for 80% of the transactions.

In Figure 5, the numbers at the bottom of the figure are the *percent of goal* numbers multiplied by the response time of 2 seconds. To get the actual time, the transaction distribution buckets are accumulated from the first bucket up to the bucket where the percentage of the defined goal is reached. The corresponding transaction time of this bucket defines the *actual time* as follows:

- Adding all the transactions completed, buckets 1 through 14 in the *transaction count* row: 140
- 80% of the transactions: $140 \times 0.8 = 112$
- Adding up buckets 1-n until we get a transaction count of at least 112, we get to bucket 7, representing an actual time of 2.2 seconds.
- The performance index = actual time divided by goal time. That means $2.2 \text{ divided by } 2.0 = 1.1$.
- A PI of more than 1 means the goal is not achieved.

2.1.1.4 RMF Response Time Report

RMF Version 5 Release 1.0 exploits the response time distribution in several reports. In the Workload Activity report, the buckets are summarized to make up a four lines histogram. RMF Monitor III has a new report that specifically exploits these buckets. Other reports also use these statistics.

2.1.2 Dispatching Priorities

A *dispatching priority* is a number from 0 to 255 associated with address space units of work (TCBs and SRBs). It indicates the priority of these units of work for getting CPU service. In goal mode, dispatching priorities are not assigned by the installation, but by SRM.

Some unique aspects of dispatching priorities in goal mode are as follows:

- All address spaces of the same service class period have the same base dispatching priority (with the exception of a mean-time-to-wait group).
- The dispatching order of all address spaces with the same dispatching priority in the same or in different service class periods are periodically rotated.
- The time slicing function previously available through controls in the IPS is no longer available in goal mode.
- Based on the PI values and the reason for delay (CPU for example), a service class period can be selected as a CPU receiver, and service class periods can be selected as CPU donors. As a consequence, the policy adjustment routine changes their relative dispatching priorities.
- The mean-time-to-wait (MTW) algorithm is only used for service class periods with a discretionary goal. There are a set of dispatching priorities at the bottom of the range that only apply to MTW discretionary address spaces. These priorities are used to rank the I/O and CPU “boundedness” as is done in compatibility mode.
- Goal mode implements I/O priority queueing exactly as would be accomplished with an IPS specification of IOQ=PRTY. Therefore, if a goal is not being achieved and the dispatching priority is raised to address that, then the I/O queueing priority is increased as well.

The compatibility mode external parameter DP (in IPS) should not be confused with the service definition *importance* keyword. They have distinct functions. For example, a high importance service class period does not imply a high dispatching priority. The *importance* value is used to determine which service class period should be a receiver of resource when there are more than one service class periods that are not meeting their goals. The external construct that in goal mode looks the most like the compatibility mode DP parameter is *velocity*, where indirectly the installation indicates the importance of obtaining both CPU and storage.

2.1.3 Target Multiprogramming Levels

The domain constructs used in compatibility mode for swapping control are replaced in goal mode by service class periods.

Service class periods have two target multiprogramming levels:

MPL in_target The number of address spaces required to be swapped-in in order to make the service class period meet its goal. This number is dynamically modified by policy adjustment routines.

MPL out_target The upper limit of address spaces allowed to be swapped-in without causing too much of a constraint to the central storage resource. It can be decreased to guarantee memory for the swapped-in address spaces. This number is dynamically decreased by the resource adjustment routine when the system

is over-utilized. Refer to 2.4.4, “System Over-Utilized” on page 52 to have more information about this situation. This number is also dynamically set by the policy adjustment routine when trying to reach the service class period goals.

2.1.4 Swap Protect Time

Prior to goal mode SRM, an installation defined a range for a system-wide “think time” to specify an amount of time that address spaces would be allowed to stay logically swapped in central storage. Until this time expired, the address spaces could not be moved to AUX or expanded storage, and so would not incur any swap delay when they became ready.

This swap protect time algorithm is a replacement for the compatibility mode logical swap algorithm using a “system think time.”

In goal mode, swap protect time is the time in milliseconds a swapped-out address space remains in processor storage (central plus expanded) before becoming a candidate for AUX. It is a value set specifically for each service class period by SRM.

The swap protect time guarantees storage to service class periods by allowing address spaces to stay swapped-out in processor storage for a certain amount of time hoping that they become ready before being moved to AUX.

2.1.5 Storage Targets

SRM controls the use of storage through storage targets. These targets look very much like the ones defined in compatibility mode by the SRM component working set manager (WSM).

Storage targets are set by the policy adjustment and resource adjustment routines. The central storage targets already existed in SRM since MVS/ESA Version 4 Release 2.0 for managed address spaces. Here, the major difference is that in compatibility mode they apply to central storage only, and in goal mode we have both central and expanded storage targets.

In general, some of the storage targets (the protective targets) keep the system from taking frames from address spaces, while other storage targets (the restrictive targets) tell the system which address spaces to take frames from first.

There are four types of storage targets:

- Protective Processor** A storage target to protect a target number of frames in processor storage. Do not migrate below target.
- Restrictive Processor** A storage target to preferentially migrate down to target. It is greater or equal to the protective target.
- Protective Central** A storage target to protect a target number of frames in central storage. Do not steal below target.
- Restrictive Central** A storage target to preferentially steal down to target. It is greater or equal to the protective target.

Storage targets are applied to each individual address space. However, service class periods for trivial transactions, such as TSO first period, have a *single* protective processor target for all the address spaces because there is not

sufficient time for a customized evaluation. Service class periods for non-trivial work may have different storage targets for each address space in the service class.

2.1.6 Expanded Storage Policy

This policy is controlled by the resource adjustment routine which decides where pages stolen from central storage should be sent — to the expanded storage (ES) or to AUX. It is a replacement for the user-controlled criteria age values in compatibility mode.

There are four categories of pages which concern this policy:

- Swap working set pages. These are the pages that are going to be automatically brought to central storage at swap-in.
- VIO pages.
- Standard hiperspace pages, pages written to a standard HS.
- Stolen pages and swap trim pages (the ones stolen at swap-out).

There are three settings for ES policy:

Protected Direct pages to ES, and protect a specific number of pages in ES.

LRU Direct pages to ES, and allow pages to be migrated in LRU order.

Space Available Direct pages to AUX unless there is significant space available on ES.

The granularity of the controls is at the service class period level, but some address spaces may have an individual ES policy where the:

- Address space has a velocity goal.
- Address space has a discretionary goal.
- Address space is a server in a dynamic server internal service class (DISC). Refer to 2.2.1, “Work Manager Services” on page 19 for more information on DISC.
- The address space has a long response time goal (> 20 seconds).

All address spaces for trivial transactions, such as TSO first period, have a constant service class-period wide policy and are managed together.

2.1.7 Capping

Capping is a technique used to enforce resource group CPU service rate maximums.

The dispatcher divides each second of elapsed time into 64 time slices. Units of work from address spaces belonging to a resource group are made non-dispatchable during some time slices in order to reduce access to the CPU to enforce the resource group maximum. The time slice where an address space is non-dispatchable is called a *cap slice*.

Capping delay is one of the delay states recorded by the WLM during sampling, and they are reported with the other delay states in the RMF report.

The granularity of the control is the resource group level and not at service class level. It means that all address spaces in all service class periods in a resource group are controlled by the same number of cap slices. Meanwhile, the dispatching priority assigned to each service class period will still be based on the goals. Therefore, work in the resource group with a more stringent goal will be more likely to run when the group is not capped.

The CPU service rate (in order to see if the limit is being exceeded) is per resource group. In a resource group formed by several service class periods, SRM enforces that the *total* CPU service rate is not going to be above the maximum.

Refer to 2.3.2.10, “Process Resource Groups above Maximum (Capping)” on page 48 for more information on capping.

2.1.8 Server Topology

WLM services provide an interface used by subsystems such as CICS and IMS to inform WLM about the current response time values for their transactions. It is through these services that WLM establishes the bond between the subsystem address spaces and the transactions service classes.

One of the sets of WLM services is called work manager services. Refer to 2.2, “WLM Service Routines” on page 18 for more information. The basic premise of work manager services is to allow WLM and SRM to meet the user-defined goals for classes of CICS or IMS transactions by managing the address spaces that serve those transactions.

WLM algorithms recognize that these transactions could be processed in more than one address space, by more than one subsystem work manager (CICS and IMS for example), in more than one MVS system image. The WLM also recognizes that address spaces can be serving a set of transactions classified to different service classes than those of the address spaces. In this case, the goals for the server address spaces themselves are then no longer honored. Instead, the servers are managed according to the goals of the transactions they are serving. In other words, any execution velocity goal declared by the installation to the server address spaces’ service classes is only going to be used at startup time, before WLM detects that the address space is a server.

An example of this process would be the WLM managing a set of CICS regions, say a TOR, several AORs and a DBCTL region. Determining which of these address spaces are processing transactions for which service class, or “who is serving whom,” is called determining the *server topology*. As mentioned earlier, the data needed for establishing such a relationship is provided by the server subsystems through the use of work manager services and sampling. For more information on server topology, refer to 2.2.4, “Server Topology Assessment” on page 27.

2.1.9 Donors and Receivers Determination

In order to meet the goals and service rate group requirements, the policy adjustment routine determines:

- Which service class periods demand more resources, that is, highest sysplex PI and local PI above a certain less-than-one threshold. They are called *receivers*.

- Resource bottlenecks (delays) in the receivers.
- *Donors* to donate these resources to receivers.

The only resources that can be donated and received are processor storage and CPU.

The donor and receiver are rarely a single address space. WLM usually refers to a receiver service class period, not a receiver address space.

WLM also ensures that the reallocation of resources from the donors to the receivers will do more good than harm. The terms used are the *net value* or the *net receiver value*, which is the algebraic sum of the projected benefit to the receiver and the projected harm to the donor.

2.2 WLM Service Routines

WLM service routines introduce concepts that other components (policy adjustment and resource adjustment) depend upon. However the most “glamorous,” and perhaps more important are the policy and resource adjustments. So you may optionally go ahead to 2.3, “Policy Adjustment” on page 34 and return here when you feel appropriate.

WLM service routines are usually used to enable WLM to cooperate with problem mode callers, such as a subsystem work manager (CICS/ESA Version 4 Release 1.0 and IMS/ESA Version 5 Release 1.0). This cooperation is used to achieve installation-defined goals for work, and to provide meaningful feedback on how well WLM has reached those goals. These services are used by CICS, IMS subsystems, and by performance monitors such as RMF. All the services are requests coming from these products to WLM.

For more detailed information on the WLM services, see the IBM SRL *MVS/ESA Programming: Workload Management Services*.

WLM services are divided into three sets of services:

- *Work manager services*: used by subsystem work manager for providing information to the WLM, such as transaction start time and transaction response time.
- *Execution delay monitoring services*, used by subsystem work manager for providing more complex information such as performance block (PB) delay states. Refer to 2.2.2, “Execution Delay Monitoring Services” on page 20 to get more information about performance blocks.
- *Workload reporting services*: used by performance monitor reporters, such as RMF, to get SRM data.

The set of workload management services to be used by the subsystem work manager depends on:

- Functions required by the subsystem work manager
- Number of the subsystem work manager address spaces
- Where these address spaces are located:
 - In the same MVS image
 - Distributed sysplex-wide

- Distributed over the network

Usually subsystem work managers are multiple address spaces formed by:

Router address space	A transaction owning region (TOR) in a CICS multi-region option (MRO), or a control region (CR) in IMS.
Execution address space	An application owning region (AOR) in a CICS MRO or a message processing region (MPR) in IMS.
Supporting address space	A file owning region (FOR) for CICS/VSAM, DL1, and DB2.

Actually all the CICS address spaces are started the same; the characteristics of being a TOR or an AOR are determined dynamically.

Within the capacity of the system, SRM manages server address spaces by allocating resources (CPU and memory) to these servers in an attempt to meet the goals of the service classes of the transactions that they serve.

WLM ignores the customer-specified service class for a server when it detects that the server address space starts to serve transactions. This is discovered either when sampling determines that the server is making use of performance blocks or by detecting that the server is uses *Report* (a Work Manager service) or *Notify* (a Execution Delay Monitoring service) service routines. WLM recognizes this type of server address space from its behavior, not requiring any direct customer specification.

2.2.1 Work Manager Services

These services allow subsystems work managers (such as CICS and IMS) to inform SRM about:

- The start of a transaction in order to get its service class from the classification rules (through the *classify* service).
- The end of a transaction in order to inform SRM about its response time (through the *Report* service).
- Which address spaces are involved in processing transactions (through the *Connect* service). This type of information is going to be used to determine the Server Topology (refer to 2.2.4, “Server Topology Assessment” on page 27).

2.2.1.1 Work Manager Services Interfaces

All the following interfaces are described in *MVS/ESA Programming: Workload Management Services*. Later we will provide some practical examples covering these interfaces. The following Work Manager services are some of the most commonly used:

Connect IWMCONN — This service allows a supervisor or PSW Key 0-7 caller such as a subsystem work manager (CICS TOR or IMS Control Region) to connect to WLM services, so that WLM may provide authorization access to other services. A token is returned to be used by subsequent services. It is used at subsystem work manager startup. The PSW key of the caller is passed, to be used in future requests.

Obtain policy IWMPQRY — This service allows the caller to get the current service policy.

Obtain service class goal IWMWQRY — This service allows the caller to get the service class goal. This service is used by CPSM.

Work classification IWMCLSFY — This is a service where a problem, supervisor, or any-PSW key caller as a subsystem work manager passes external *work qualifiers*, such as name, class, user ID, account information, LU name, network ID, or “product-specific” attribute, to WLM. Then, WLM compares them with the customer’s classification rules and associates the transaction with an “external” service class. It is used at transaction start processing.

Response time reporting IWMRPT — This service allows WLM to obtain from the subsystem work manager the total response time for a transaction (by reporting the arrival time). This response time information is used to update the distribution of response times for the associated service class in the response time buckets, and to recalculate the performance index for that service class. It is issued at transaction-end processing, and if there is a PB associated with the transaction, it indicates:

- The end of the begin-to-end (BTE) phase.
- That the PB should no longer be associated with the just-completed transaction.

Refer to 2.2.2, “Execution Delay Monitoring Services” to get more information on PBs and BTE phases.

Only normal completions are included in the response time information, so the performance data is not skewed by abnormal completions. This service is also used in the server topology to recognize that an address space is a subsystem work manager server.

Disconnect IWMDISC — This service is used by the subsystem work manager (any CICS address space) to disconnect from the WLM.

2.2.2 Execution Delay Monitoring Services

These are a collection of services required by subsystem work manager in order to keep the internal delay states of a set of transactions belonging to a served service class. They are used for reporting purposes: for example, to report the work flowing from a subsystem work manager to a data manager.

The states are kept in performance blocks (PBs). The following are some properties of PBs:

- Created and deleted by the subsystem work managers.
- Another name for a PB is *Monitoring environment*.
- Can be allocated in common or in a private area.
- PBs are associated with transactions and not with service classes.
- It is possible that one transaction may be associated with more than one PB at same time.
- PBs do not contain the transaction response time.
- PBs are used to:

- Assist SRM in determining the Server Topology.
- Report internal subsystem work manager states and delays through a bit map. There are no counters.
- The state and delay bits in a PB are set by the change state service (IWMMCHST) issued by a subsystem work manager.
- SRM samples the PB every 250 milliseconds. The results are collected by a workload reporting service (IWRMCOLL) and presented to a performance monitor such as RMF.
- SRM does not try to fix the delays presented in the PBs; they are internal subsystem work manager delays.
- There are two types of PBs:
 - PBs associated with a begin-to-end phase (BTE), allocated, for example, by a CICS TOR.
 - PBs associated with an execution phase (EXE), allocated by a CICS AOR, CICS FOR, IMS/MPR and DL1.
- The states recorded by all CICS FORs and all CICS AORs are merged together when RMF reports information on the CICS execution phase.

It is important for a CICS specialist to have a report where he (or she) can obtain detailed information about the states and the delay events which occurred when transactions belonging to an external service class were running. For this purpose the lifetime of a transaction is segmented into phases.

2.2.2.1 Begin-to-End (BTE) Phase

The BTE phase begins when the transaction starts to be processed in a TOR and ends when the TOR receives it back from an AOR and finishes off. Strictly speaking, the BTE phase starts when the TOR PB is initialized (by the initialize PB service) and ends when the TOR PB is reported (by the report service). The BTE phase duration is roughly the response time minus the TOR queue time, VTAM processing and network time.

The BTE is physically represented by a PB. This PB is defined, initialized, and associated with an arriving transaction by the TOR. During the BTE phase, the TOR (through the change state service) sets states and delay bits in the PB. These bits reflect what is going on with the transaction. For example, when the TOR passes the transaction to an AOR, the “delay_for_conversation” bit is set.

As mentioned previously, SRM samples the BTE PB every 250 milliseconds. The results are collected by a workload reporting service (IWRMCOLL) issued by RMF and reported in the Workload Activity report in a line called CICS BTE.

Please refer to Chapter 3, “Workload Manager Performance Monitoring” on page 59 to see RMF reports displaying information about BTE and EXEC phases.

Note: The IMS CR does not define a PB for a BTE phase.

2.2.2.2 Execution (EXEC) Phase

An EXEC phase has details about different steps in the transaction's life that are not described in the BTE phase.

There are two types of EXEC phases:

EXEC1 This phase is when the execution is done by a *transaction execution* address space (an AOR or FOR, or an IMS/MPR).

EXEC2 This phase is when the execution is done by the *supporting* (database) address space, such as a copy of DBCtl at the correct software level.

The **EXEC1 phase** is **not** continuous, it:

- Begins when the transaction starts to be processed by an AOR. A PB is initialized.
- Is interrupted when the DL1 address space is processing the transaction.
- Is resumed when the AOR receives the transaction back. A transfer return service was called.
- Is finished when the TOR is finally invoked. A notify service was called.

EXEC1 is roughly the AOR/MPR transaction service time.

Strictly speaking, the EXEC1 phase starts when the AOR PB is initialized (by the initialize PB service) and ends when AOR finishes transaction processing (by the notify service).

The EXEC1 phase is physically represented by a PB. This PB is defined, initialized, and associated with a transaction by an AOR. Therefore, during the AOR execution, the same transaction has two PBs: one defined in the TOR, reporting for the BTE phase (with the delay-for-conversation bit on) and the other defined by the AOR reporting the EXEC1 phase. SRM samples both PBs every 250 milliseconds. This implies that all the EXEC1 phase is included in the BTE phase.

During the EXEC1 phase, the AOR (through the change state service) sets states and delay bits in the PB. These bits reflect what is going on with the transaction during AOR execution. For example, when the AOR passes the transaction to DL1 and DL1 has not yet started processing the transaction, the "delay_for_other_product" bit is set.

Note: IMS/MPR defines a PB for an EXEC1 phase. All CICS transactions that only execute in the TOR do not have an EXEC phase.

The **EXEC2 phase** is continuous. It begins when the transaction starts to be processed by DL1 and finishes upon return to the AOR. EXEC2 duration is roughly the DL1 transaction service time.

Strictly speaking, the EXEC2 phase starts when the DL1 PB is transferred (by the transferred continue services) by DL1 and ends when DL1 transfers it back (by the transfer return service).

As with the other phases, the EXEC2 phase is physically represented by a PB. This PB is defined, initialized (by the relate service) and associated with the transaction by DL1.

It is important to note that the transfer service issued by DL1 declares that the AOR PB is no longer meaningful. Therefore, during the DL1 execution, the same transaction has three PBs: one defined in a TOR, reporting for the BTE phase (with the delay-for-conversation bit still on), one defined in the AOR for the EXE phase (but temporarily not meaningful) and the last defined by DL1, reporting the EXEC2 phase. SRM samples the meaningful PBs every 250 milliseconds. The EXEC2 phase does not overlap with the EXEC1 phase.

During the EXEC2 phase, DL1 (through the change state service) sets state and delay bits in the PB. These bits reflect what is going on with the transaction during DL1 execution. For example, when DL1 detects a lock held situation, the "delay_for_lock" bit is set.

Note: DB2 and CICS FOR (VSAM) do not define an EXEC2 phase PB.

In Figure 6, there is one example of one BTE phase (CICS TOR) and just one EXEC phase (CICS AOR and FOR merged):

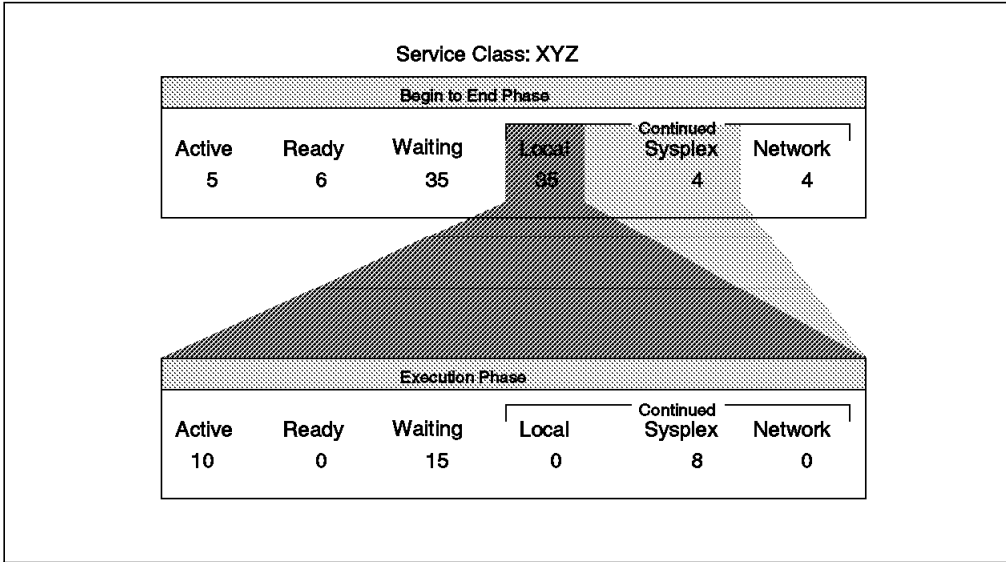


Figure 6. BTE and EXEC Phase

In Figure 7 on page 24, there is another example with one BTE phase (CICS TOR) and two EXEC phases (CICS AOR and DL1):

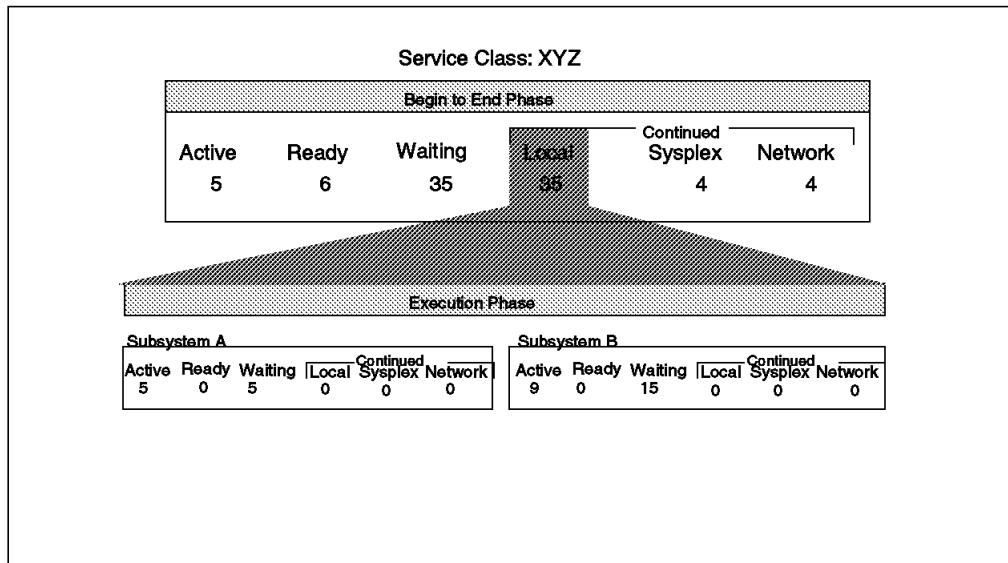


Figure 7. BTE and EXEC Phases

2.2.3 Performance Block States

The following are some of the PBs state bits. These states provide the CICS or IMS view of the transaction.

Active The transaction has an associated program which is being executed by the subsystem scheduler.

Ready The transaction has an associated program which is ready to run, but not being dispatched by the subsystem scheduler.

Idle A conversational CICS transaction waiting for user think time in TOR. This is seen in the BTE phase.

Delay by Lock The internal subsystem delay for enqueues.

Delay by I/O Waiting for I/O operations and related functions.

Delay by Conversation Waiting for a conversation. For a CICS BTE phase, this is the delay experienced by a TOR after sending the transaction to an AOR. For a CICS EXE phase, this is the delay experienced by an AOR after sending the transaction to an FOR.

Delay by Distributed request Waiting for a distributed request; that is, some CICS TOR is connected to an AOR by a network connection.

Delay by Local Wait for local conversation,

Delay by Sysplex Waiting for a conversation with another MVS image somewhere in the sysplex.

Delay by Remote Waiting for a conversation through a network connection to a system not in the sysplex.

Delay by Timer Waiting because the application voluntarily requested a delay for a certain amount of time.

Delay by Product Delayed due to the invocation of another product. An example could be a CICS AOR calling DL1. This delay starts when the AOR issues change state service (IWMMCHST) until the moment when DL1 issues the transfer service (IWMMXFER).

Delay by Miscellaneous This means being delayed due to any other reason not described by the previous delays. For CICS, this is usually caused by the application waiting on a user defined ECB.

Note: TSO, BATCH, APPC and OMVS do not have PBs because of the close relationship between the address space and the transaction concept. The SRM will sample existing control blocks, not PB state bits to determine the delays for CPU and storage.

2.2.3.1 Execution Delay Monitoring Services Interfaces

All the following interfaces are described in *MVS/ESA Programming: Workload Management Services*.

Some of the most used interfaces to invoke execution delay monitoring services are:

Create IWMMCREA — This service establishes a token to be associated with a PB in order to keep track of the subsystem work manager execution delays encountered by an incoming transaction. Each IMS Message Processing Region creates one PB. There is one PB allocated for each potential CICS task. This number is the user-specified MAXTASK parameter plus a few extra. Create also defines the PSW Key which the subsystem work manager runs under later.

PB initialization IWMMINIT — This service supplies WLM with information referring to the new transaction (Service Class, user ID, LU name, etc.). This data is stored in the PB. IWMMINIT is issued when:

- A transaction arrives in the TOR, defining the start of the *Begin-to-End*(BTE) phase
- The transaction is moved to an execution address space (AOR), defining the start of the *Execution* (EXE) phase. In this case, the subsystem work manager may provide, through this service, the time stamp when the transaction arrived in the execution address space. It is used to calculate the transaction execution time (a subset of the transaction response time).

When the transaction is processed by a support address space (for example, DL1) the relate service, not the initialization service is used.

PB Relate IWMMRELA — This service with the Function=Create initializes a PB (like IWMMINIT) relating this PB with another PB associated with the same transaction. SRM samples both PBs but the ones in the related PB represent the current state of the transaction. This service is used by a database manager (named *dependent environment*), such as DL1, after receiving a transaction from a transaction execution (named *parent environment*), such as the CICS AOR. A token must be passed by the parent to the dependent in order to invoke the service. Parent and dependent must be (for this service) in the same MVS image.

The PB relate service with the Function=Delete disassociates the parent and dependent PBs.

Transfer IWMMXFER — This service with Function=Continue is issued by a database manager after receiving control from a transaction manager. It indicates that the real state of the transaction now resides in the database manager's PB. To use this service, the

database manager and the transaction manager must be in the same MVS image.

Switch IWMMSWCH — This service with Function=Continue is identical to the IWMMXFER service, but the continuation of the transaction processing is “somewhere else” (same image, in the sysplex or in the network). RMF shows under Switched Time % (Local, SYSPL, Remote) the percentage of the response time that the transaction stayed switched.

The switch service with Function=Return indicates the end of the transaction switched mode.

Transfer IWMMXFER — This service with Function=Return resets the dependent PB to free. At this point WLM/SRM recognizes that the dependent PB no longer represents the transaction. However, the parent and dependent PBs are still related for any future transfers, such as with a second call to the database manager. It also reports that the execution delays for a transaction are now to be determined from the prior PB associated with the transaction.

Change state IWMMCHST — This service is used to inform WLM about the current transaction state, such as:

- Ready
- Active
- Being delayed by reasons such as:
 - I/O
 - Locks
 - Conversation
 - Other products, for example, a transaction manager waiting for a database manager

All these changes are reported by setting bits in the PB which may be seen by WLM later during sampling.

Notify IWMMNTFY — This service is used to inform WLM/SRM that the transaction execution phase (EXEC1) and the execution time are over. This execution time starts to be measured when the transaction is moved to an AOR and ends at notify WLM service call. Refer to Figure 13 on page 42 to see the difference between execution time and EXEC phase time. The number of notifies occurring in the system is reported by RMF in Workload Activity report in the field EXECUTD. An execution phase could be split among several subsystems, and the delays associated with each distinct subsystem are reported separately. The performance monitor should combine all information by service class by subsystem to provide a sysplex view. This service is also used in the server topology to recognize that an address space is a subsystem work manager.

Delete IWMMDELE — This service is used to delete the PB.

2.2.4 Server Topology Assessment

When all address spaces start, WLM views them as nonservers. Later an address space is recognized as being a server if:

- It is associated with a performance block that represents a transaction. The association was done via execution delay monitoring services.
- The subsystem address space issues either a report (IWMRPT) or notify (IWMMNTFY) for a transaction.

In order to evaluate the server topology, WLM must first understand which external service classes are being served by each server. This is accomplished by WLM maintaining a served class history for each server address space.

Customers define service classes for transactions to be processed by server address spaces. These service classes are referred to as “external service classes” (ESCs), and they have real customer-defined goals. WLM groups some work into *internal* service classes. Some internal service classes are called *dynamic* when they are associated with server address spaces (such as CICS) serving the transactions with the external service classes. These dynamic internal server classes (DISC) are used to manage to the external service classes goals. For example, let’s assume that the WLM wants to help a given service class (ESC) representing some CICS transactions. The WLM must know which address spaces must be given more resource because the CICS transactions in that ESC run in multiple address spaces. *A DISC is the set of address spaces that serve a given ESC*, and it is the set of address spaces in the DISC that will be given the resources.

DISCs have special names defined internally by WLM as *\$\$SRMSnnn*. The number of *\$\$SRMSnnn* service classes depends on how many external service classes are served and by how many combinations of server address spaces.

Server topology is the function used to create the DISC and map it to the external service classes (ESC) representing transactions with goals.

Each DISC contains a set of server address spaces that are serving the transactions belonging to the same set of external service classes. All of these address spaces will be managed together.

Each recognized server address space has counters indicating how many times it is detected serving each ESC. As stated previously, the incrementing of the counters is done by either PB sampling, or report and notify processing.

Refer to Figure 8 on page 29 for an example of a server topology. The DISCs on the left are the set of address spaces that were found to be providing service to the ESCs on the right.

2.2.4.1 How the Server Topology is Determined

The following is an example of the methodology used to determine the server topology.

Let’s imagine that address space server 1 was found servicing class A 150 times, class B 150 times, and class C 200 times. Address space server 2 was found servicing class B 100 times, class C 100 times, class D 200 times and class E 100 times and so on. All the service classes belong to the same resource group. The following list shows who is serving whom:

server 1 = {A,B,C}
server 2 = {B,C,D,E}
server 3 = {A,B}
server 4 = {A,B}
server 5 = {F}

To find the minimum number of dynamic internal service classes, it must be determined for each server address space set all other server address space sets which are equal (that is, those address spaces that serve the same set of external service classes, and which belongs to the same resource group). For the above example, this exercise would result in the following dynamic internal service classes being built:

Dynamic Internal Service Class W = {A,B,C} formed by address space server 1
Dynamic Internal Service Class X = {B,C,D,E} formed by address space server 2
Dynamic Internal Service Class Y = {A,B} formed by address space servers 3 and 4
Dynamic Internal Service Class Z = {F} formed by address space server 5

The dynamic internal service classes (DISC) are used as accumulators of data for managing servers. As the same address space can be serving multiple external service classes, the counts are used to determine a weighting factor of not only who is serving whom, but also how much.

The topology is reassessed and the DISCs potentially rebuilt once per minute, based on a full performance block sampling for topology or on the topology deduced from report and notify calls.

Refer to Figure 8 on page 29 for an example of a server topology. There is an ESC for each service class representing the CICS transactions. If the WLM wants to help the ESC for CICS Tran Type 2 in this figure, then the only address space that can be helped is the DISC representing the TOR. Similarly, if WLM wanted to help CICS Tran Type 4, then the weighting factors would be used to determine whether to help the TOR address space, or the set of address spaces for AOR1 and FOR.

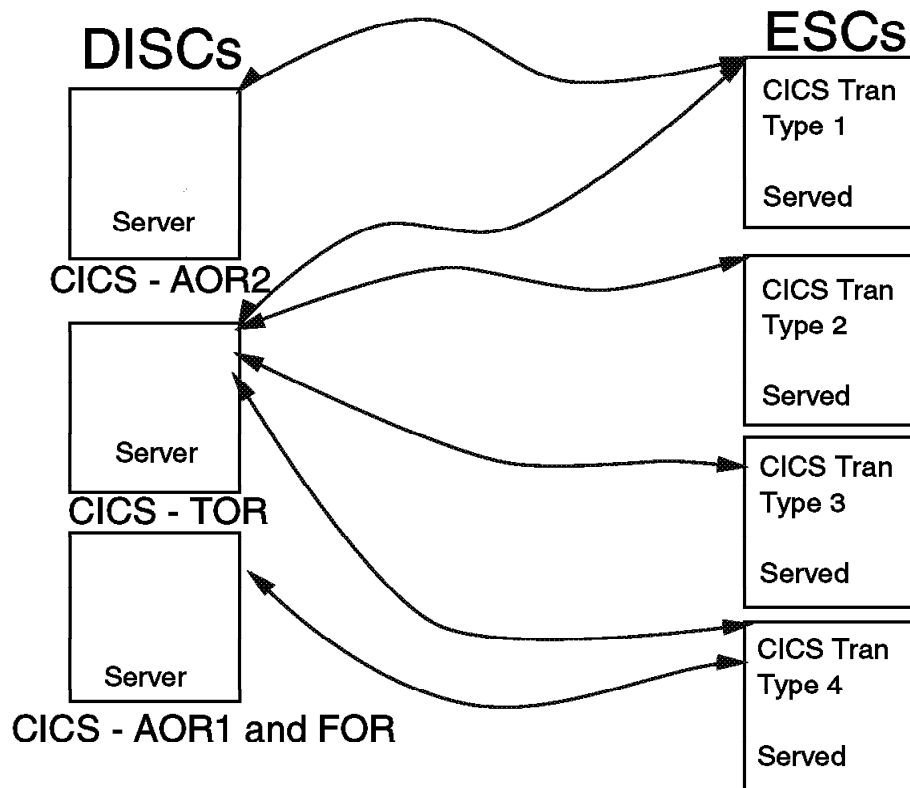


Figure 8. Server Topology Example

2.2.5 WLM Services / IMS Example

This example shows a simplified chronological sequence of services encompassing work manager, execution delay monitoring services and IMS:

- At IMS Control Region Startup:
 - Connect service (IWMCONN) is passed a list of ASIDs to establish a set of server address spaces. This list includes DBRC and DLISAS in addition to the IMS Control Region.
- At IMS Message Processing Region Startup:
 - Create service (IWMMCREA) returns a token that is associated with the PB. There is just one PB per each MPR because there can be only one transaction in an MPR at one time.
- At IMS/CR transaction start:
 - Work classification service (IWMCLSFY) specifies the service class. Schedule the program in an MPR. Note that there is no PB for the transaction in the Control Region.
- At IMS/MPR transaction execution:
 - PB initialization service (IWMMINIT) supplies WLM with a PB for this transaction. This is the start of the EXEC phase. IMS does not have the begin-to-end (BTE) phase because IMS/CR does not use a PB.

- Notify service (IWMMNTFY), informs WLM/SRM that the transaction execution phase is over. Returns control to IMS/CR and enters the wait state.
- At IMS/CR transaction execution:
 - Response time reporting service (IWMRPT), that allows WLM to obtain from the subsystem work manager the total response time. It indicates the end of the transaction.
- At IMS/MPR Termination:
 - The delete service (IWMMDELE), deletes the PB.
- At IMS/CR Termination:
 - Disconnect service (IWMDISC).

2.2.6 WLM Services CICS / CPSM / DL1 Example

This example shows a simplified chronological sequence of WLM services calls encompassing work manager services, execution delay monitoring services and CICS/CPSM/DL1. Figure 9 on page 33 can be used to follow the example. It shows only the calls associated with the life of one transaction. The WLM service calls issued at startup time and at subsystem termination time are not shown. In the figure the length of time intervals (in percentage of the response time) between calls is also shown. The figures should match the part of RMF Workload Activity report presented in Figure 10 on page 34 which follows the events of this example:

- At TOR Startup:
 - Connect service (IWMCONN) connects the CICS system to the WLM and returns a token which is used in all other service calls.
 - Create service (IWMMCREA) returns a token that will be associated with a TOR PB. The number of PBs is determined by Maxtasks plus a few more for the system CICS transactions.
- At AOR Startup:
 - Connect service (IWMCONN) connects the CICS system to the WLM and returns a token which is used in all other service calls.
 - Create service (IWMMCREA) returns a token that will be associated with an AOR PB.
- At DBCtl Startup:
 - Create service (IWMMCREA) returns a token that will be associated with a DL1 PB.
- At the start of a transaction controlled by CPSM:

If this transaction is for a service class that CPSM hasn't seen before, CPSM will obtain the service class goal with the IWMWQRY service and save the value in a table for subsequent transactions. The CPSM code running as the CICS Dynamic routing exit then recommends the "best" AOR.

CPSM has two algorithms for routing transactions.

- In the CPSM GOAL algorithm:
 - CPSM makes a list of AORs which are eligible to run the transaction. AORs which are down or are not "healthy" are removed from the list. The "health" of an AOR is determined by *CPSM agent transactions*

running in the AOR. These transactions continuously monitor the AOR to ensure that transactions are flowing.

CPSM calculates the average response time for this transaction code delivered by each AOR and uses this average to calculate a performance index for this AOR.

The list of AORs is sorted based on the response time delivered by each AOR and the goal for this transaction's service class.

An AOR is chosen that will keep the number of AORs running this service class to a minimum.

CPSM does not understand Percentile goals. However, to make sure the service class goals table is kept current, CPSM listens for the Event Notification Facility (ENF) signal which WLM raises when a policy change occurs and refreshes the contents of its service class goals table.

One consideration for setting up classification rules is that CPSM is a real CICS region and runs some transactions. Unless specifically handled, these transactions could be classified in service classes other than expected.

- In the QUEUE algorithm, CPSM always tries to select the AOR providing the minimum queue time and does not use the services provided in goal mode.
- At TOR transaction execution:
 - Work classification service (IWMCLSFY) obtains the service class.
 - PB initialization service (IWMMINIT) supplies WLM with a PB for this transaction. This is the start of the BTE phase.
 - Change state service (IWMMCHST), for example from active to ready. The change state service is a fast, inline service causing a minimum of overhead. Bits in the PB are set by this service for the sampling routines to detect. More than one change state may happen in sequence. The last one (before invoking AOR) is `waiting_for_conversation`.
 - Switch service (IWMMSWCH) says the continuation of the transaction processing is "somewhere else" (same image, in the sysplex or in the network). This implies that execution delays for this transaction are recorded in another PB. Both PBs will be sampled by SRM. If the switching by CICS is by MRO, then the target AOR must either be in the same MVS image or in the sysplex.
 - Send the transaction to a CPSM selected AOR.
- At AOR transaction execution:
 - PB initialization service (IWMMINIT) supplies WLM with a PB to reflect the AOR's view of this transaction. This is the start of the EXEC phase.
 - Change state service (IWMMCHST) informs that the DL1 address space is being invoked. More than one change in the state may happen in sequence.

Many change states may be issued during execution, with `ready_to_run`, `active`, `waiting` being a typical sequence.

Invoke DL1.
- At DL1 transaction execution:

- The PB relate service (IWMMRELA) with the Function=Create, relates the DL1 PB (dependent) with the AOR PB (parent).
- The transfer service (IWMMXFER) with Function=Continue, indicates that the real state of the transaction now resides in the database manager's PB (for future change state and sampling purposes).
- Change state service (IWMMCHST) informs that the DL1 address space has a status change (PB bit is changed accordingly). The state changes from active to ready and vice versa. More than one change in the state may happen in sequence.
- The transfer service (IWMMXFER) with Function=Return, indicates that the real state of the transaction is no longer recorded in the DL1 PB. However the PBs are still related for future transfers.
- The PB relate service (IWMMRELA) with the Function=Delete, breaks the relationship of the DL1 PB (dependent) with the AOR PB (parent). Return control to AOR.
- At AOR transaction execution:
 - Change state service (IWMMCHST) informs that control is back in the AOR, clearing the bit indicating the transaction is "switched."
 - Notify service (IWMMNTFY), informs WLM/SRM that the transaction execution phase is over. Returns control to TOR and looks for work.
- At TOR transaction execution:
 - Switch return (IWMMSWCH)
 - Response time reporting service (IWMRPT), that reports the total response time to WLM. This indicates the end of the BTE phase. The component that does classify must do the report.
- At DL1 completion:
 - The PB relate service (IWMMRELA) with the Function=Delete, disassociates the parent and dependent PBs.
 - The delete service (IWMMDELE), deletes the PB.
- At AOR completion:
 - The delete service (IWMMDELE), deletes the PB.
 - Disconnect service (IWMDISC).
- At TOR Completion:
 - The delete service (IWMMDELE), deletes the PB.
 - Disconnect service (IWMDISC).

Reporting Phases

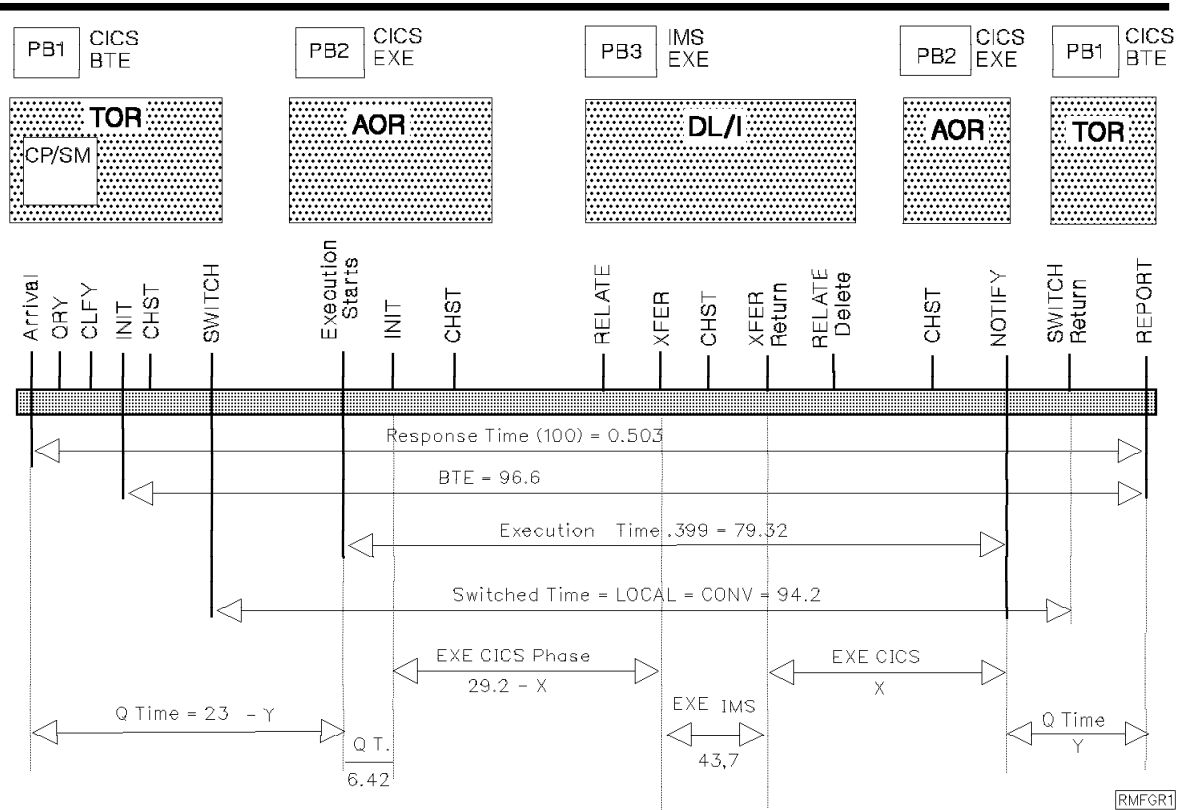


Figure 9. Timetable of Events. Q Time is the AOR queue time. It is derived by the subtraction of the EXEC phases from the Execution Time.

The following Figure 10 on page 34 is a fragment from an RMF report which show how the various states of the transactions is reported by RMF.

REPORT BY: policy=WSTPOL02 WORKLOAD=CICS SERVICE CLASS=CICS RESO

-TRANSACTIONS-- TRANSACTION TIME HHH.MM.SS.TTT
AVG 0.00 ACTUAL 000.00.00.503
MPL 0.00 QUEUED 000.00.00.117
ENDED 579 EXECUTION 000.00.00.399
END/SEC 0.95 STANDARD DEVIATION 000.00.00.847
#SWAPS 0
EXECUTD 559

-----RESPONSE TIME BREAKDOWN IN PERCENTAGE-----
SUB P TOTAL ACTV RDY IDLE -----WAITING FOR-----
TYPE LOCK I/O CONV DIST LOC SYSPL REMOT TIME PROD MISC
CICS BTE 96.6 2.4 0.1 0.0 0.0 0.0 94.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0
CICS EXE 29.2 14.2 4.1 0.0 0.0 4.0 0.0 0.0 0.0 0.0 0.0 0.0 6.9 0.0
IMS EXE 43.7 40.3 0.0 0.0 3.4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

-----STATE-----
SWITCHED TIME (%)
LOCAL SYSPL REMOT
CICS 94.2 0.0 0.0
CICS 0.0 0.0 0.0
IMS 0.0 0.0 0.0

Figure 10. Some RMF Data

2.2.7 Workload Reporting Services

Workload reporting services provides information for performance monitors to report on how well an installation is meeting its performance goals. The services provide information about work that is processed by many address spaces, and also view this data by transaction. The most used interface is the IWMRCOLL that retrieves all workload activity data. It is used by RMF to build the workload activity report in goal mode.

2.3 Policy Adjustment

It is the SRM component in charge of response, or its purpose is to meet the transaction's service class goals and resource group service rate requirements as stated in the service policy. This target is achieved by the distribution of resources among the service classes. Throughout this process some trade-offs need to be addressed and the general golden rule is equalize PIs by importance within resource group constraints. In MVS/ESA Version 5 Release 1 the policy adjustment routine is invoked every 10 seconds.

2.3.1 Donors and Receivers Concept

In order to meet the goals and requirements, the policy routine determines:

- The service class period that most needs help meeting the goals in the policy is called a receiver.
- Which resource bottlenecks (delays) are affecting the receiver. It is important to mention the distinction between a service class receiver and a service class that needs help. A service class receiver has to do with an unmet goal. The receiver is the most important work that needs help meeting its goal. A service class that needs help has to do with detected

delays. A service class may be missing its goal, but if the delays are for things that the SRM does not control, such as I/O or locks, then the service class can not be helped, and will not be a receiver.

- Which donors donate resource to the receiver.

The only delay states that are used by SRM are CPU_delay, Stor_delay, MPL_delay and SWIN_delay; the only resources that can be donated or received are processor storage and CPU.

The donor and receiver are sometimes a single AS, but more often they are all the ASs in a service class period. We usually refer to a receiver service class period, not a receiver address space.

WLM uses its plots to project a recipient benefit and a donor cost. The net benefit must be positive before a resource adjustment will take place.

In order to derive the net value of receiver/donor relationship SRM creates the following plots to track how well work is being processed:

- System paging delay plots
- Period MPL delay plot
- Period ready user average plot
- Period swap delay plot
- Period paging rate plot
- Period proportional aggregate speed plot
- Address space paging plot

This idea of plotting data started in SRM Version 4, Release 2 with the working set manager (WSM) component, plotting the address space paging characteristic curve and the system paging curve.

For workloads containing subsystem transactions such as CICS and IMS, donors and receivers can be classified as:

- Goal (receivers or donors) when its service class period is served by another (a server) service class period. This service class period is called *served service class*, or an *external service class*. These service classes are for CICS and IMS transactions where there may not be a one-to-one mapping of a transaction to an address space:
 - One transaction can be run in multiple address spaces.
 - One address space may run transactions from multiple served classes.
- Resource (receivers or donors) - also called servers - when its service class period is a *server service class* of another service class period. The *Resource* SC is the receiver or donor of resources which affects the performance of the *goal* service class. Resources are allocated to the server (or *internal service class*). The server's service class delay state samples are apportioned to the served service classes based on the time the server was serving transactions from the served service class. These delay samples are used to find the bottlenecks.

2.3.2 Policy Adjustment Loop

This loop looks very much what system programmers do when MVS Version 5 Release 1 is running in compatibility mode and the performance is not good:

- If they cannot buy more resources then they create the illusion that they bought some by not wasting resources or by using the discretionary resources.
- They steal resources from the least important donor work and give them to receivers.

The following shows the flow executed every time that the policy adjustment routine is invoked. The major objective is to find *one* receiver to be helped. Refer to Figure 11.

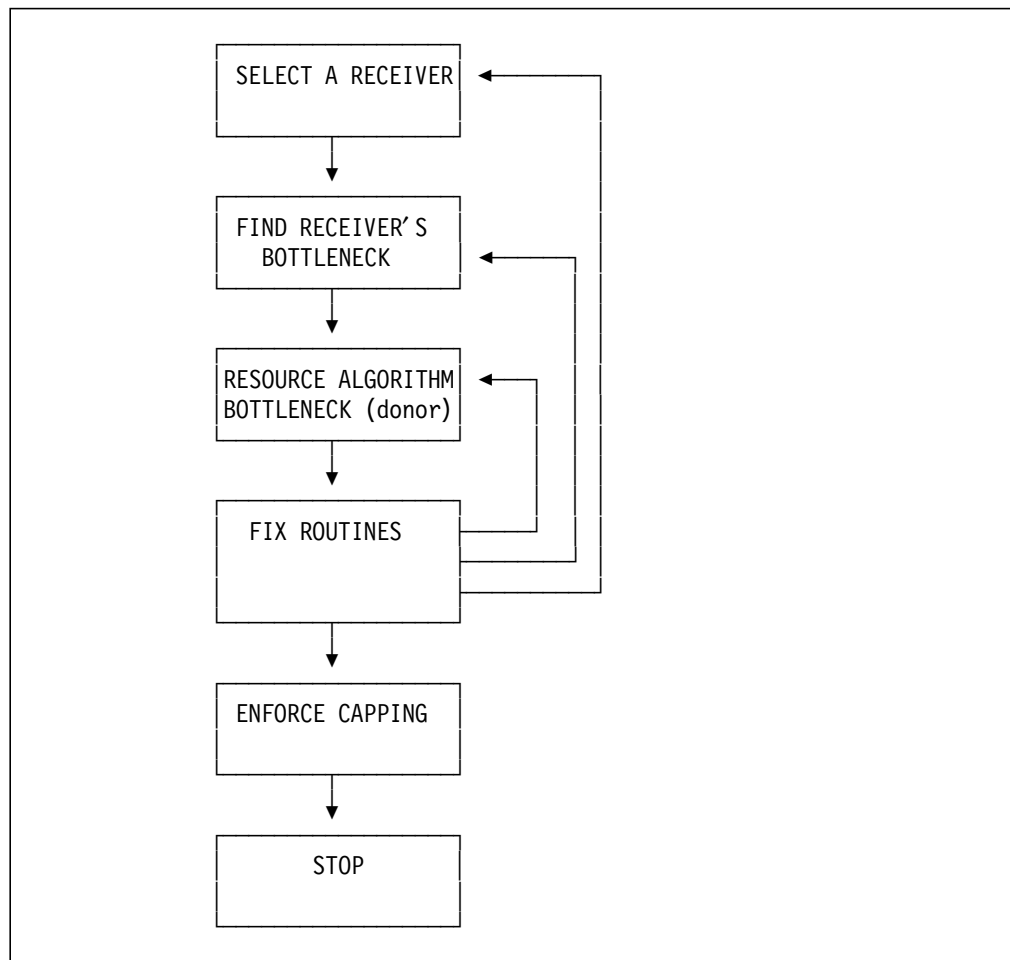


Figure 11. Loop Summary

Note: All the actions executed in the policy adjustment loop are logged every 10 seconds in a SMF 99 record. These records can help in analyzing specific events, however, SMF 99 recording can consume a lot of DASD space and is not recommended for normal operation. For general reporting and tuning information for a system in goal mode, you can use SMF 72 records (RMF Workload Management report).

2.3.2.1 Select a Receiver

The following describes the four criteria that can be checked to select a candidate to be a receiver. These criteria were covered more generally in a previous topic in this book, nevertheless here are more details. If one of the criteria is met then the search stops as it is the aim of the routine to find just one receiver at each invocation. This is the order in which the evaluation takes place:

1. A service class period that is not meeting its goal running in a resource group which is running below the resource group service rate minimum. The best receiver is chosen by largest amount below resource group service minimum, by most importance, by highest PI.
2. A service class period not meeting its goal running in resource groups running between resource group service minimum and maximum (ok groups). The best receiver is chosen by most importance and highest PI within importance.
3. A discretionary period (universal donor) with resource group service below minimum. The best receiver chosen by largest amount below resource group service minimum. This case is the only one that a service class period with a discretionary goal (PI = 0.81) can be selected as receiver.
4. A service class period meeting its goal. The best receiver chosen by largest PI.

Sometimes a resource group is skipped even when the CPU consumption is below the minimum because the analysis of the local data in every MVS image indicates that another image can do better than this one. This avoids all the SRMs from overreacting to such a situation.

If a receiver service class (ESC) is associated with CICS address spaces (DISC) running important transactions with a PI greater than one, together with less important CICS transactions, then resources will be given to the DISC to help the important ESC. Obviously the less important ESCs will get the benefit of these resources. This has been referred to as giving the low importance work a “free ride.”

Note: Referring to server service classes (DISC) and served external service classes (ESC), only the served external service classes (ESC) are selected as receivers, that is the ones that have goals (goal period). An internal service class (DISC) cannot be selected. (Refer to 2.2.4, “Server Topology Assessment” on page 27, to get more information about server topology and DISC and ESC concept.)

2.3.2.2 Find a Receiver’s Bottleneck

Generally speaking in order to find a bottleneck in the receiver, the SRM controlled general resource delays are analyzed:

- Processor delay
- Paging delay
- Swap page-in delay
- MPL (OUTR) delay

If there is no SRM controlled resource being a bottleneck (could be I/O, capping, database calls, locks or other WAIT), then the receiver cannot be helped, and the

algorithm will select another receiver candidate by going back to the *Select a Receiver* step.

There are two cases:

- **Non-Server Case**

The primary way to help a non-served receiver is to alleviate one of the receiver's delay reasons. The receiver's largest non-cross memory delay and the receiver's largest cross memory delay are analyzed. The larger of the two delays is returned and the *Individual Resource Algorithm per Bottleneck* step is invoked. If a cross-memory delay is returned, then the target address space's OUCB address is also returned. This is so the cross memory fix routine can help the target address space, which in turn should help the receiver.

- **Server Case**

In order to find a bottleneck for a receiver served service class (ESC) also called "goal period" the following steps are executed:

- Use the topology to understand the proportion of time (weights) each server DISC spent serving the chosen receiver ESC.
- Calculate the delay samples for the receiver ESC.

The receiver ESC is not associated specifically with address spaces and only address spaces can be sampled when looking for delay reasons. To address it the delay samples of the server DISC address spaces are apportioned to the receiver ESC based on the previous calculated weights. In other words, the delay samples for the server DISC address spaces are scaled by the proportion of time the server DISC was serving this receiver ESC in relation to the amount of time the server period was serving any DISC.

- Find the resource with the largest delay in the receiver's (ESC). Select the related DISC. This is the service class period passed to the fix routines.

2.3.2.3 Individual Resource Algorithm per Bottleneck

Individual resource algorithm selects donors for a receiver. The donation is related with the resource being a bottleneck for the receiver. The individual resource algorithm is invoked for each bottleneck in the receiver. The following is its flow:

1. Select donors (by the reverse order of their PIs) and actions (as donating storage or CPU)

There are four criteria to find a donor:

- Service class periods meeting goals (including discretionary in resource groups with service above minimum); best donor chosen by lowest PI.
- Discretionary periods below resource group service minimum; best donor chosen by smallest amount below resource group service minimum.
- Service class periods not meeting goals in resource groups running between minimum and maximum; best donor chosen by least importance, by lowest PI.

- Service class periods not meeting goals in resource groups running below minimum; best donor chosen by smallest amount below resource group service minimum, by least importance, by smallest PI.

As you see importance is only a criterion when goals are not met.

2. Assess *receiver value*.

A policy action is not taken to help a service class period unless there is sufficient receiver value. An action must have sufficient value to the receiver. If this were not done the policy code could continually take action to help one service class period that has little effect, and ignoring other service class periods that are less important, but not meeting their goals.

3. Access impact to donors (net value). A receiver can receive resources from donors or discretionary. If while analyzing a donor, the projected harm to the donor is more than projected improvements to the receiver, the condition is traced (in SMF record 99) and another donor is selected. The “net value assessment” considers all external service policy specifications: resource group service minimums and maximums, importance and goals. Also projected PIs are going to be derived.
4. Loop through steps 1 on page 38 to step 3 for additional donors and actions as required.
5. Adjust policy for resource and trace (SMF 99).

The resources to help the receiver may also come out of what is referred as “discretionary resources”, those that can be reallocated with little or no effect on the system’s ability to meet performance goals. An example of a discretionary resource is the amount of central storage occupied by swapped-out address spaces protected inside of the swap protect time.

2.3.2.4 Fix Routines

The Fix routines are in charge of fixing the bottleneck by giving a specific resource from the donor(s) to the receiver. There is one of such routines per every tracked delay resource:

- Auxiliary Storage
- Multiprogramming level (MPL)
- Swap-out
- Processor

2.3.2.5 Fix_AUX_Storage_Delays Routine

This routine is invoked if a receiver service class period is experiencing its largest delay because the address spaces within the period are being delayed due to pages are ending up on AUX DASD memory, as:

- Private area paging
- Common area paging
- Cross memory paging
- VIO paging
- Standard paging

This routine addresses these paging delays through the use of the protective processor storage target (refer to 2.1.5, “Storage Targets” on page 15 for more details). This target protects an address space from losing processor storage

due to UIC steal, physical swap-outs (working set and trimmed pages), expanded storage migration and consequently it is less delayed by page-ins from auxiliary storage.

The fix routine has three cases:

- Private area paging for short response time goal period (trivial TSO)
- Private area paging for long response time goal period
- Common and cross memory paging delays

Fix_Aux_Storage_Delays for Short Response: Let us start our analysis with the ***Private Short Response*** case.

For a trivial TSO service class period, it is assumed that most of the paging delays are caused by the trim at swap out (logically or physically). Also, in this type of service class period the transactions do not stay around long enough for SRM to gather sufficient data to manage the working sets of the address spaces individually. Therefore, decisions to change the storage protective processor storage of these address spaces are made on a service class period-wide basis. Every address space in the service class period has the same processor storage protective target.

To project the effect of changing the period wide protective storage target the period wide paging plot is used. The x axis is the average processor storage size of address spaces that left the service class period in the last 10 seconds. The y axis is the average number of AUX page-ins (scaled by 10) done by address spaces that left the period in the last 10 seconds. Refer to Figure 12 on page 41.

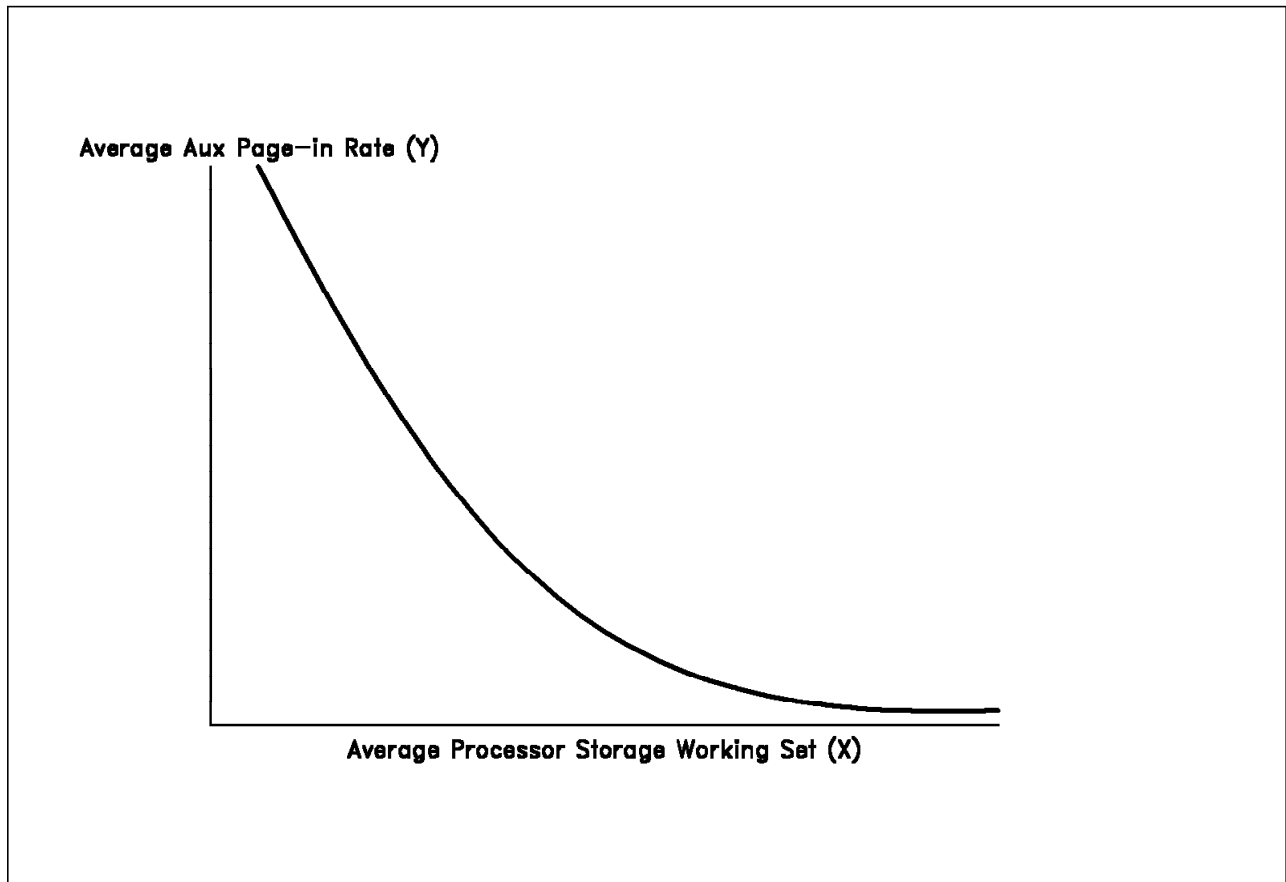


Figure 12. Period Wide Paging Plot

This plot is used to project the change to the periods AUX page-in rate caused by modifying the period's protective processor storage target. The projected page-in rate change is used to calculate the change in the AUX delay samples that the service class period will experience in the next interval. From these samples the response time or velocity deltas and performance index delta are calculated for the service class period. The performance index delta is used to determine if the trade-off being considered is a good one.

The following are some SMF 99 trace codes connected to this routine:

- PA_INC_PSI_TAR - Receiver's target was increased
- PA_PSI_NA_NET_VAL - target not increased due to insufficient processor storage
- PA_PSI_NA_REC_VAL - Not enough receiver value in the action
- I_TAR_UNAB - target was not increased because current target not absorbed

Fix Aux Storage Delays for Long Response: Let now start the analysis of the *Private Long Response* case.

When a period has a goal (other than a short response time goal) and experiences private area paging delays, a different approach is used. There are two reasons for this difference:

- Transaction in such a period should stay around long enough for SRM to learn how the transaction is using processor storage.

- Transactions in such a period are probably going to use processor storage very differently, so that a “one size fits all” approach to solving paging delays may not work. Instead the individual address space’s working set is managed to reduce paging delays.

In this case the action taken to address AUX paging delays is to increase the individual protective processor storage target of one address space in the service class period. The address space chosen to help is the one that can have the biggest impact on the paging overhead that address spaces in the service class period are experiencing.

The individual address space paging plot is used to project the effect of these changes. The x axis of this plot is the address space processor storage size. The plot has 3 y curves:

- Page-in rate from AUX
- The CPU cost of paging from AUX measured in terms of the CPU cost of paging to/from AUX in milliseconds divided by the residency time in seconds
- The productive CPU rate which is the address space TCB + SRB time measured in milliseconds divided by its residency time measured in seconds

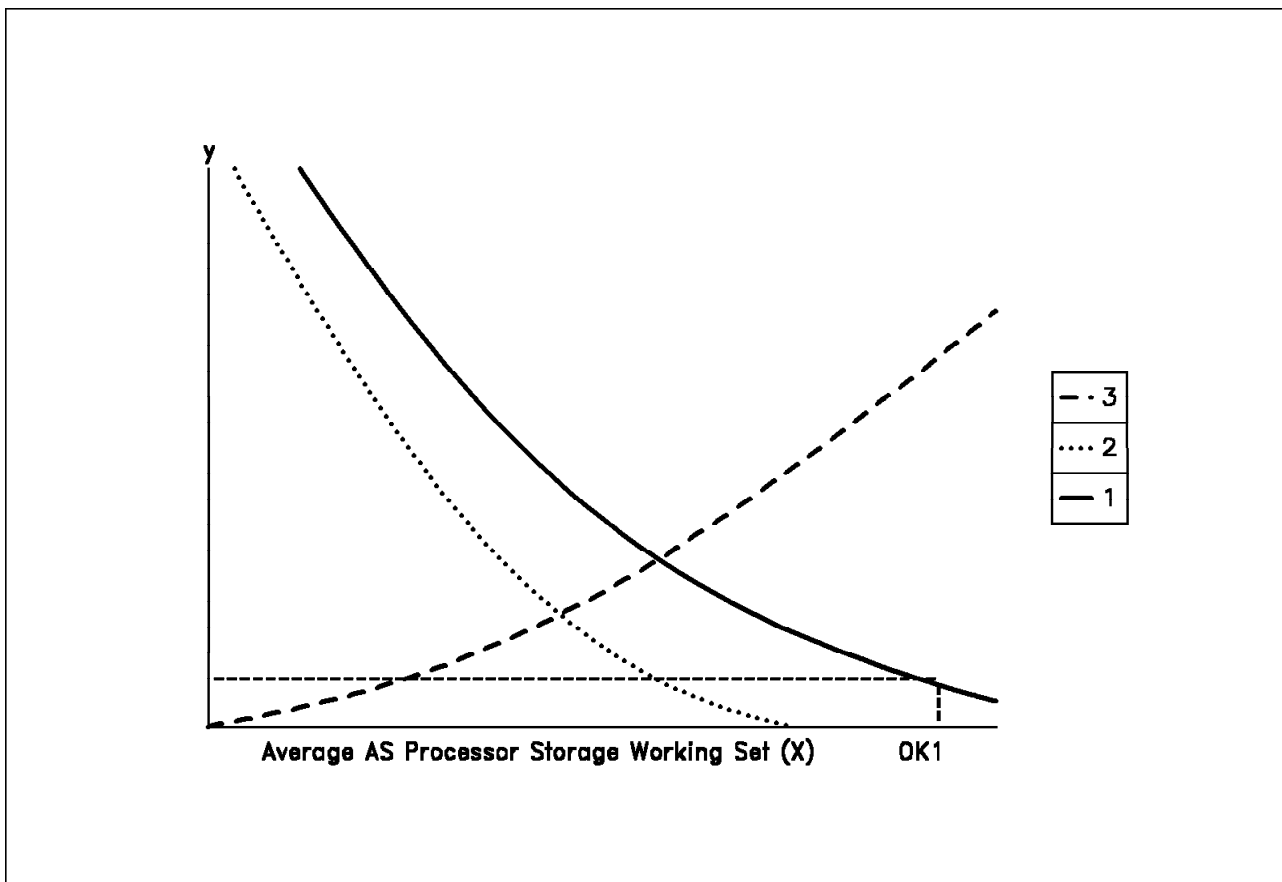


Figure 13. Address Space Paging Plot

Please refer to Figure 13.

This plot is used to project the change in the AUX page-in rate for the address space caused by modifying the address space’s protective processor storage target. The change in the AUX page-in rate is then used to calculate the change

in AUX delay samples which in turn can be used to get the performance index delta in much the same way as the short response time goal case.

The following are some SMF 99 actions associated with this algorithm:

- PA_INC_ISI_TAR - The address spaces individual storage target was increased.
- PA_ISI_NA_NET_VAL - target was not increased because FIND_STORAGE could not find sufficient processor storage.
- PA_ISI_NA_REC_VAL _ target was not increased because of insufficient receiver value.

Note that multiple PA_ISI_NA_NET_VAL and PA_ISI_NA_REC_VAL traces can be issued for the receiver because each net and receiver value failure is traced.

Fix Aux Storage Delays for Common and Cross Memory: Let's now start the analysis of the ***Common and Cross Memory Paging*** case.

The approach taken for common storage delays is very simple. No plot is kept to make projections. Instead the common storage protective target is set to the smaller of:

- One megabyte above common current size or
- 10% above its current size.

The assumption is made that this increase gets rid of all the paging delay samples that the receiver is currently seeing. The total common area delay samples for the receiver are fed into calculations similar to those described above to get a PI for the receiver. If this action has sufficient receiver value and SRM can find enough processor storage, the change to common target is made. Cross memory delays are handled in much the same way as delays for common storage by increasing the processor storage protective target in the address space which is suffering the page faults.

2.3.2.6 MPL_Delay_Fix Routine

This routine is invoked if a receiver service class period is experiencing its largest delay because the address spaces within the period are being Multiprogramming Level (MPL) delayed.

The MPL of a period is the number of address spaces in that period that are swapped-in in central storage at any given time. An address space in a period experiences MPL delays when that address space is swapped-out and is ready to be swapped in, but SRM has not (or cannot) increase the MPL in the period to allow the address space to be swapped in. In order for MPL_delay_fix routine to alleviate a period's MPL delay, it must allow the period's "swapped-out and ready" users into central storage. It is done by increasing the period's MPL in_target by one and maybe the period's MPL out_target by one as well. Refer to 2.1.3, "Target Multiprogramming Levels" on page 14 to have a quick look in these two targets.

On the other hand, WLM must "find" and "ensure" that there is enough central storage available to contain another address space typical of that period. This is done through the invocation of the find_storage routine.

Let us see some definitions useful for understanding this routine:

MPL Slot A logical entity representing a potential swapped-in-and-ready user.

MPL in_target The number of MPL slots representing the number of address spaces in a period that must be swapped-in (i.e. in central storage) in order for the period to meet its goals.

MPL out_target The maximum number of MPL slots representing the number of address spaces in the period that could be allowed in the swapped-in state (that is, in central storage) without causing too much of a constraint to the central storage resource.

Ready User Average The average number of users in the period that are ready to run (not in long wait, not in detected wait, not in terminal wait). These address spaces are either out-and-ready or in-and-ready. It is typical of this number to fluctuate greatly between policy adjustment intervals.

Long Term Ready User Average The long term average number of users in the period that are ready to run. These address spaces are either out-and-ready or in-and-ready.

Current MPL The current number of address spaces in a period that are swapped-in-and ready.

See Figure 14 and Figure 15 on page 45 for examples of the plots used to predict the effect of an MPL change.

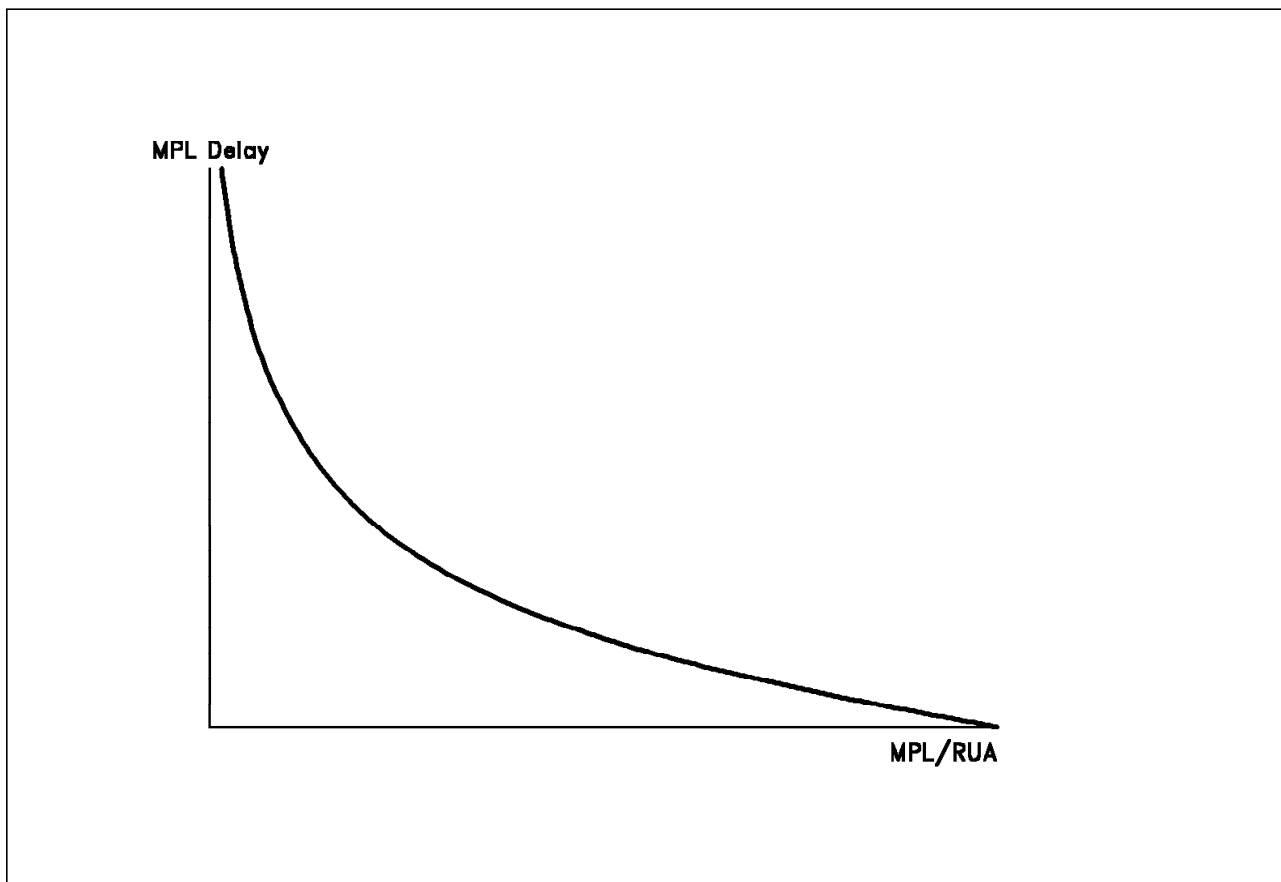


Figure 14. Period MPL Delay Plot

The plot is designed to show how response time may potentially be improved or degraded by changing MPL slots.

- X value: the percentage of ready users who have an MPL slot available to them. This is the ratio of MPL available to the period and the number of ready users in the period. Another way to look at this number is the ratio of how much MPL has been given to how much is needed.

$$X = \frac{\max(\text{in_target}, \text{Current MPL})}{\max(\text{RUA}, \text{Long Term RUA})}$$

- Y value: the MPL delay per period completion (in milliseconds). This is a measure of time from when a user was out-and-ready.

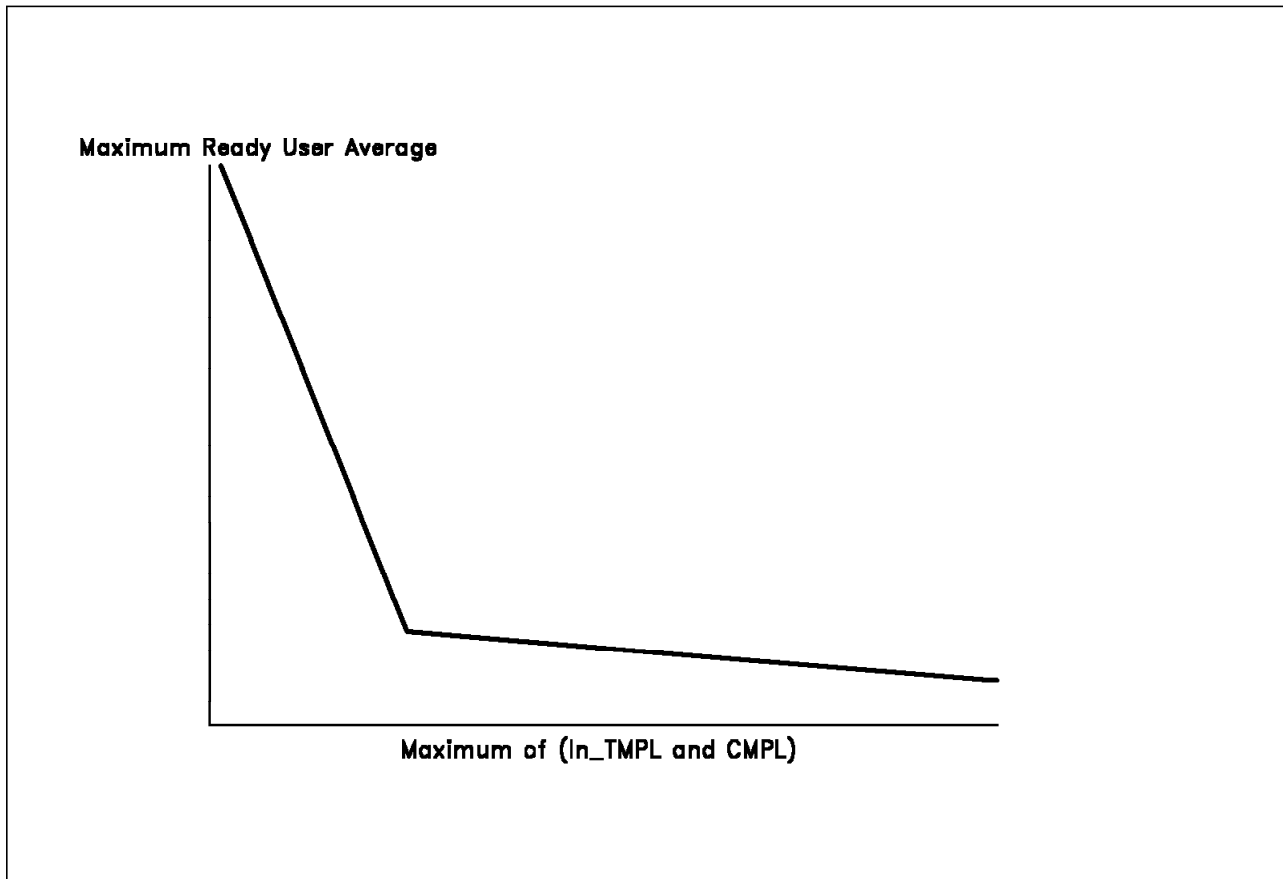


Figure 15. Period Ready User Average Plot

The plot is designed to predict the number of ready users when given a particular MPL in_target for a period. Also it can be used to avoid lowering a period's MPL in_target to a point that causes out-and-ready-users to experience large increases.

- X-value: the number of MPL slots (times 16) available to the service class period, or the larger of the MPL in_target and the current MPL.
- Y-value: the maximum number of ready users (times 16) averaged over a two second interval.

In an online environment if transactions are severely delayed (by MPL for example) then the number of ready transactions grows. (Transactions are not ending but new transactions of other users are arriving.) With the same number of MPL slots, the number of swapped out-ready is going to increase.

A sharp “knee” is expected in the curve of this plot (a little exaggerated in Figure 15). The thought here is that if a period were given enough MPL, the number of ready users in that period would remain fairly constant. The number of arriving transactions is on average equal to the number of departing transactions. This is indicated by the flat, horizontal section of the curve. The “knee” of the curve indicates the point of MPL sensitivity. That is, the point at which if not enough MPL slots are allocated to the period, the number of out-and-ready users starts to increase. The “knee” in this plot becomes especially apparent when the ready user average plot is associated with a period for TSO short transactions, which are very sensitive to MPL.

Now let us analyze the process of assessing the action. The basic premise of the MPL assess routine is to assess the potential effects on a period if the period’s MPL targets is increased by one.

The logic of the MPL assessment is as follows:

1. Use the MPL delay plot to determine the MPL delay that the period is currently seeing (Y-value).
2. Read the MPL delay plot to determine the “Delay that the period would probably see,” if given the MPL slots proposed (Y-value). This is done by calculating the current “ratio of (MPL slots available plus MPL slots proposed)” and Ready User Average and then reading the Y-value.
3. The delta between “MPL delay that the period is currently seeing” obtained from the first reading and the “MPL delay that the period would probably see” obtained from the second reading is considered the amount of MPL delay that could be addressed if the period were given the MPL slots proposed. From this delta MPL delay value (which would be the delta in the response time) the MPL assessment routine is able to determine the delta PI.
4. This delta PI is then used to calculate the projected local PI and the projected sysplex PI. Based on these projected PIs a receiver value is then calculated to determine if the proposed action of giving the period another MPL slot has enough value to the receiving period.

Trace Codes: The following trace codes are associated with the MPL fix routines.

PA_INC_MPL - Increase the period’s MPL targets

PA_MPL_NA_NET_VAL - No action taken due to lack of net value (that is, not enough central storage found)

PA_MPL_NA_REC_VAL - No action taken due to lack of receiver value

2.3.2.7 Swap_Delay_Fix routine

This routine is invoked if a receiver service class period is experiencing its largest delay because the address spaces within the period are being delayed due to page-in activities from AUX due to a swap-in process. The solution is to increase the *swap protect time* of the service class period.

Refer to 2.1.4, “Swap Protect Time” on page 15 to get more information about this time.

The following concepts are used by swap_delay_fix routine in order to decide the new value of the swap protect time:

Average time_swapped_in_ps. The average time between when an address space is logically swapped-out and the time it is either moved to AUX or swapped back in. This time is kept in OUCB and in the swap_delay_history table for the corresponding service class period. This table is composed of 33 buckets containing the frequency of address spaces per time range. (From zero to 88 seconds).

Average swap_delay time. The average time between when an address space swap-in starts and the swap-in finishes is called the address space's swap_delay time. This time is kept in OUCB and in the swap_delay_history table of the corresponding service class period. This time is not obtained by the SRM general sampling routine. It is a real measured value.

Also a service class period swap_delay plot is built in order to derive the new value of swap protect time. Refer to Figure 16.

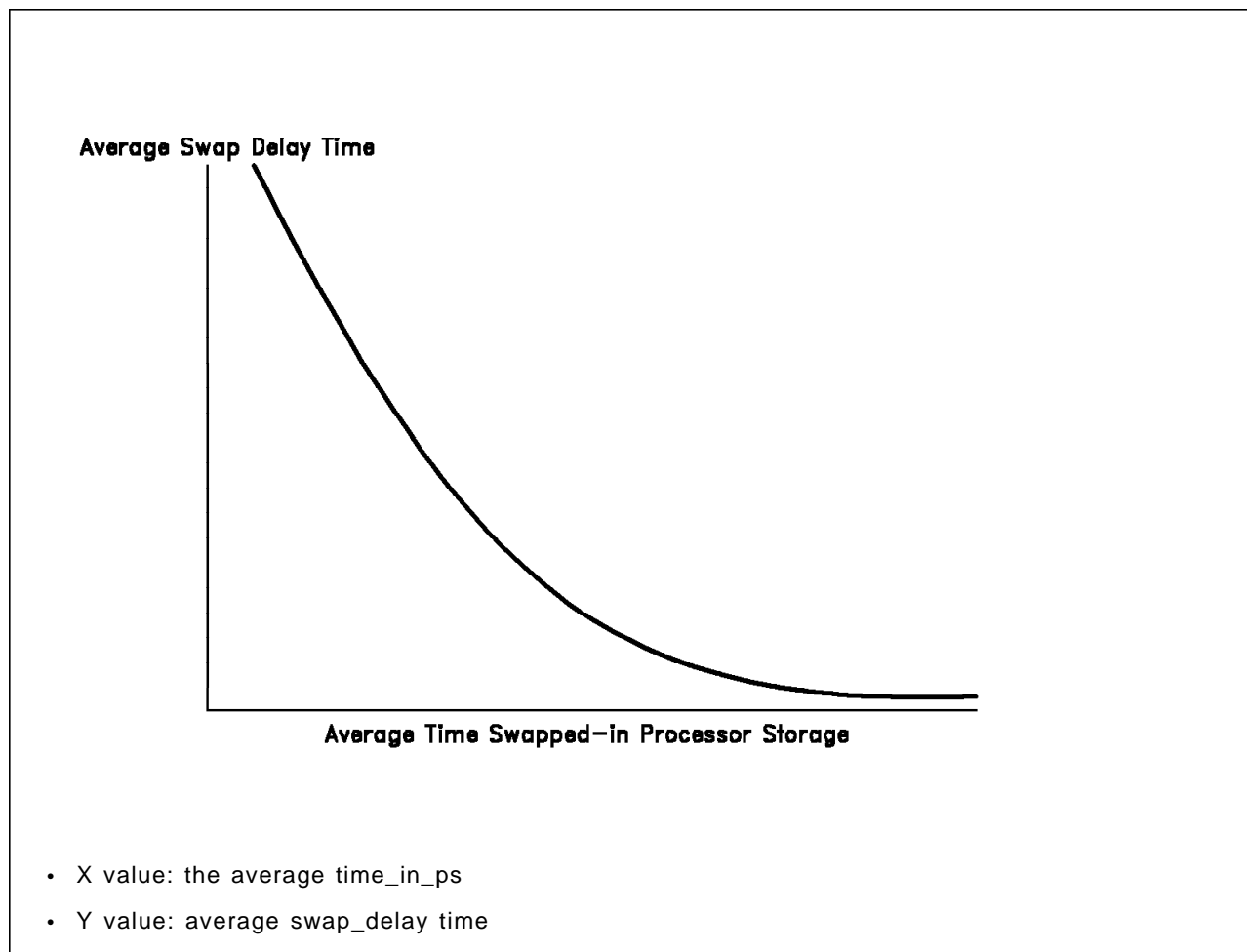


Figure 16. Period Swap Delay Plot

The plot decreases (that is, the longer that address spaces are allowed to stay swapped-in in processor storage, the smaller the amount of swap delay they have).

The period swap delay plot is used to project the change in the swap delay the service class period will experience if its swap project time is modified. The projected change in swap delay is used to calculate the change in the swap delay samples that the service class period will experience in the next interval. From these samples the response time or velocity deltas and performance index delta are calculated for the service class period. The performance index delta is used to determine if the trade-off being considered is a good one.

2.3.2.8 Processor Delay Fix

Processor delays are addressed by increasing the dispatch priority of the receiver or decreasing the dispatch priority of one or more donors or both. Projections are made for the time the affected service class periods will consume and the new wait-to-using ratios the affected service class periods will experience after the dispatching priority changes. The time is projected based on the total processor time available, the maximum demand of the service class period, and the maximum demand of work at higher and equal dispatch priorities. The wait-to-using ratios are projected using the actual observed data collected for the wait-to-using ratio at each priority adjusted for the maximum demand moved from one priority to another.

The projected times and wait-to-using ratios are used to calculate the change in processor_using and delay samples that work in the service class period will experience in the next interval. For non-served classes, these samples are used to calculate response time or velocity deltas and performance index deltas for the receiver and donors. For served classes, the samples are calculated for all servers of receivers or donors and are then apportioned to the served classes based on the relative number of observations of each server serving each class. A proportional aggregate speed is then calculated for each served receiver or donor and the performance index delta is read off the proportional aggregate speed plot. In both the served and non-served cases, the performance index deltas are then evaluated to determine if the trade-off is a good one. If there is net value to the dispatching priority tradeoff, the change is made.

2.3.2.9 Stop If One Receiver was Already Helped

If one receiver was already “helped” then stop the loop and finally execute the step of *capping*. If no receiver was helped return to the *Select a Receiver* step to look for one additional receiver.

2.3.2.10 Process Resource Groups above Maximum (Capping)

The purpose of the capping function is to control the amount of CPU service rate that units of work in a set of address spaces in a resource group consume. A resource group is a WLM service policy construct used by the installation to limit by a maximum (capping) or to guarantee by a minimum some CPU capacity to the address spaces belonging to the service class periods that constitute the resource group. These minimum and maximum capacities are measured in unweighted CPU service rate consumed in the resource group across the sysplex. A resource group can be formed by distinct service classes with different importance values.

Policy adjustment code forces this objective by measuring the CPU service rate consumed by the group (locally and in the sysplex). The CPU service rate values are accumulated first on local systems and then across the sysplex for a total. The total value of the consumed service rate is used to determine if it exceeds the resource group maximum service rate. This comparison then

determines how much to throttle address spaces in the group. This throttling is done by limiting the dispatchability of the address spaces in the resource group.

Cap Time Slices: To implement capping the elapsed time is divided into 64 time slices. Each time slice then represents 1/64th of the total time. In MVS/ESA Version 5 Release 1 each time slice represents .5 of an SRM second (with the TUNIT constant set to 2 which is the default).

Units of work from address spaces belonging to a resource group are made non-dispatchable during some time slices in order to reduce access to the CPU to enforce the resource group maximum. The time slice where address spaces in a group are set non-dispatchable is called a *cap slice*. The time slice where address spaces in a group are set dispatchable is called an *awake slice*.

Because two groups may accumulate service units at a different rate, each time slice for a group is associated with a previously measured average CPU service rate that the resource group collects. Therefore, knowing how much above the maximum a resource group is, it is simple to derive the number of cap slices the address spaces in the group are going to get.

All address spaces in a group are going to be set non-dispatchable for the same length of time.

Because the cap slices for different groups are evenly spread across the time slices and the time slices are of relatively short duration, the changes in dispatchability do not appear as abrupt service changes.

Every 10 seconds (the policy adjustment interval time) all resource groups are reappraised to determine if further adjustment is necessary. If so the times that groups are to be set dispatchable or nondispatchable are reevaluated. The 64 time slices and the cap slices are then reassigned.

Data Used in the Capping Algorithm: Input data is needed in order to enforce capping in a resource group. The inputs to the capping algorithm are the:

- Group maximum CPU service rate extracted from service policy.
- Local CPU service rate at particular importance value.

2.3.2.11 Resource Groups Maximums

Address spaces in resource groups are made non-dispatchable for a sufficient proportion of time to limit the processor time used by address spaces in the group to the maximum specified. The maximum is the total across the sysplex for all address spaces in the group. The address spaces are switched between dispatchable and non-dispatchable frequently enough so that the address spaces are only slowed, not stopped completely for extended periods.

All address spaces in the group on each system are made dispatchable for the same proportion of time and during the same intervals. Address spaces in the same group but on different systems may be made dispatchable for different proportions of time and during different intervals. The proportion of time address spaces are made dispatchable on different systems is based on the importance and quantity of work in the group running on each system. Work on the system that has more of the group's higher importance work may be made dispatchable for a larger proportion of time than lower importance work in the same group on another system. The intent is to equalize performance indexes

for service class periods of the same importance within the resource group constraints across the sysplex.

Capping delay is treated as another form of processor delay when managing dispatching priorities to meet goals. Consequently service class periods within a capped resource group may have their dispatch priorities increased to address capping delay as well as processor delay.

2.4 Resource Adjustment Routine

The resource adjustment routines are the SRM component that takes care of the throughput, keeping the resources of the system effectively utilized.

Its functions are:

- Detecting and addressing under-utilized, over-utilized and shortage conditions.
- Managing individual address space working sets through Working Set Manager (WSM).

The resource adjustment algorithms respect the constraints set by the policy adjustment algorithms, avoiding taking actions that would cause work to miss goals. In other words, goals have precedence over throughput.

Resource adjustment is done within the resource monitor-2 (RM2) routine which gains control every two seconds on larger systems. These actions are never applied to a service class period but to address spaces. A service class period is a service policy construct and has nothing to do with resource adjustment.

Note: WSM is still a major component in goal mode. The major differences between compatibility and goal mode are:

- WSM in goal mode optimizes central storage and expanded storage, in compatibility mode only central storage is optimized.
- Address spaces may also be selected for monitoring by the policy adjustment routine.

2.4.1 Concepts in Resource Adjustment Routine

The following are some of the terms and concepts used internally by the resource adjustment algorithms. Some of them are common to the policy adjustment routine and are repeated here.

Productive CPU Rate The ratio of the time an address space is doing useful work to the time it is in storage. It is defined as TCB + SRB time measured in milliseconds divided by residency time measured in seconds.

AUX Paging Cost Rate A measure of the CPU cost of paging by the address space. It is defined as the calculated (not measured) CPU cost of paging to/from AUX measured in milliseconds of CPU time over residency time.

Total Paging Cost Rate A measure of the CPU cost of paging by the address space. It is defined as the calculated (not measured) CPU cost of paging to/from expand/AUX measured in milliseconds of CPU time over residency time.

Net Productive CPU Rate A measure of how much an address space adds to running the CPUs in a system productively. It is defined as the productive CPU rate for an address space minus the paging cost for an address space.

Unmanaged CPU Paging Cost Rate CPU cost of paging done by address spaces that do not have a restrictive storage target.

Process Storage OK1 Point The amount of processor storage an address space needs to have so that it is spending less than 5% of its time paging from AUX.

Central Storage OK1 Point The amount of central storage an address space needs to have so that it is spending less than 5% of its time paging from expanded or AUX.

Monitored Address Space An address space that is having data collected to help manage its working set. Monitored address spaces have points plotted on their address space paging plot and address space central paging plot.

Managed Address Space Address space with either a restrictive central and/or expanded storage target.

Discretionary Storage Frames that can be reallocated without significantly effecting the work in the system. Included are available frames, frames containing pages with a high Unreferenced Interval Count (UIC), and any amount of frames containing pages above a restrictive target. There is a discretionary count for both central and expanded store frames.

2.4.2 Address Space Monitoring

Here we start to cover the major components of the resource adjustment component of SRM.

WSM only manages the working set of an address space if it is monitored. A monitored address space has data collected about its storage usage in the form of two plots:

- The address space “paging plot” tracks how an address space performs in relationship to the amount of processor (central + expanded) storage the address space has.
- The address space “central storage paging plot” tracks how an address space performs in relationship to the amount of central storage the address space has.

The address space paging plot is pictured in Figure 13 on page 42.

The address space central storage paging plot has the same shape as the paging plot curve and the following axes:

- X value: the average central storage working set size of the address space over the last RM2 interval.
- Y value: represents three sets of data:
 - Page-in rate from AUX and expanded storage over the last RM2
 - AUX and expanded storage paging cost rate
 - Productive CPU rate

Address spaces in periods that have short response time goals or that are in the SYSTEM or SYSSTC service classes are never monitored.

At the beginning of the RM2 interval the swapped-in address spaces are searched looking for non-monitored address spaces that should be monitored. The following are roughly the set of conditions that an address space must meet before it is considered for monitoring:

- It must have collected enough CPU time so we have some reliable data about it
- One of the following must be true:
 - The address space is paging significantly.
 - The address space is at least as large as the average monitored address space.
 - The policy adjustment code wants this address space monitored.

2.4.3 Resource Adjustment Actions

On a time-driven basis (1 SRM second) the resource monitor1 (RM1) routine takes samples. After 2 seconds the RM2 and its subroutines collect data about what happened over the last RM2 interval and then plot points on address space plots. Next, a sequential list of system conditions are verified. If one of the following conditions is present actions can be ordered and executed:

- System is over-utilized
- System unmanaged CPU paging high
- System AUX paging high
- OK1 action
- System is under-utilized
- Phase change
- Working set management

2.4.4 System Over-Utilized

The system is consider over-utilized if:

- On average a significant number of address spaces were waiting for the CPU, or
- The percentage of central storage fixed is above the RCT fixed high thresholds RCCFXTTH, and RCCXETH (as in compatibility mode).

If the system meets one of these criteria, the MPL out target of a service class period is decreased so that an address space is swapped out and demand for system resources is reduced. First it attempts to find a discretionary period to lower the MPL out target. If no discretionary period is eligible, a period with goals is chosen in donor order. The chosen period's MPL out target is lowered enough to cause one address space in the period to be swapped out. If the system is over-utilized no other action is taken during the RM2 interval, whether or not a period was found to reduce MPL.

2.4.5 System Unmanaged CPU Paging Cost High

System-unmanaged CPU paging cost is the cost of CPU due to paging done by all address spaces that do not have a restrictive storage target. It is considered high when over 10%. One reason the system might be paging heavily is that one or more large running jobs (managed or not) do not fit in available storage. In such a situation it is probably better to let these jobs page and keep the rest of the system with little paging.

Therefore resource adjustment attempts to convert some of the unmanaged paging to managed paging:

- First, try to reduce unmanaged paging by reducing the restrictive central target of an already managed address space by a value (squeeze). The idea here is to decrease the working set of a managed address space in order to decrease the unmanaged paging. If this reduction takes the address space plot curve to a high paging rate then the address space is swapped-out since it is not likely to get any useful work done with such a low target. In general the same address space is “squeezed” until the unmanaged paging problem is solved or the space is swapped.
- Secondly, if no address space can be squeezed, the next step is to try to manage a new address space. The following criteria are used in choosing an additional address space to manage.
 - An address space in a period meeting goals is chosen before one missing goals
 - The least important address space is chosen
 - The address space furthest from OK1
 - The largest address space

The address space chosen to be managed is given a restrictive central target below its current size. If no address space is chosen to be managed, a period is chosen to have its MPL reduced, discretionary first and then in donor order.

2.4.6 System Auxiliary Paging High

This situation happens if the system AUX paging rate is higher than the DASD paging subsystem can handle. The sign that the paging subsystem is overloaded is that paging in from AUX starts taking longer to process due to the queue time building up. The “system paging responsiveness plot” is used to detect this situation. Refer to Figure 17 on page 54.

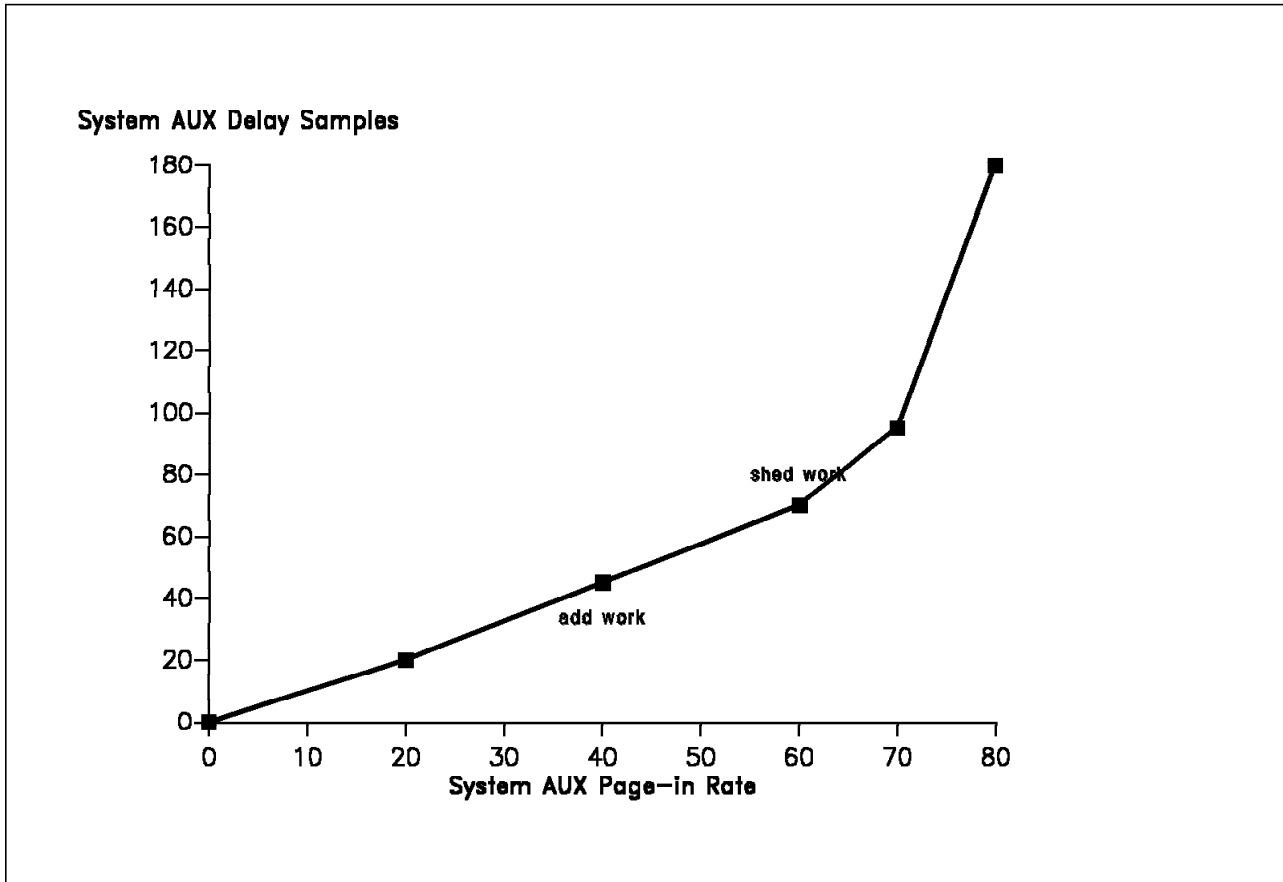


Figure 17. Paging Responsiveness

There is one of these plots per system. Points are plotted every minute. The plot has the following attributes:

- X value: the system AUX non-blocked page-in rate. This rate does not include the pages that are automatically brought in a block together with the faulting page.
- Y value: the total AUX delay samples (private, xmemory, common)

If the DASD paging subsystem is doing well, then the plot will show that AUX delay samples are growing proportionately with systems's AUX page-in rate. If the paging subsystem becomes overloaded, the plot will show AUX delay samples growing significantly faster than the systems's aux page-in rate.

The first point at which the rate of growth of AUX delay samples becomes more than twice the AUX page-in growth rate is called the "shed work point." The last point at which the rate growth in AUX delay samples is close to proportional with the AUX page-in growth rate is called the "add work point." When the current point on the plot is at or above the shed work point, a period is chosen to reduce MPL in the same fashion as the over-utilized case.

2.4.7 OK1 Action

The next condition verified is the need for executing an OK1 action. An OK1 action steadily increases the protective target of one address space, until the OK1 point is reached. The advantages of an OK1 action are:

- Less address space page-in rate
- To find the central storage OK1 point of a monitored address space. Finding an address space's OK1 point fills out the address space's central paging plot, which gives the WSM code more information to manage the space.

An address space is only given one chance to go through an OK1 action because these actions have the potential to be disruptive to the system.

When an OK1 action is started for an address space, it is given a protective central target above its current size. During each RM2 interval after that in which the OK1 action is continued the protective target increases by a target if the following conditions are true:

- Address space absorbed last target change, that is the current working set is greater or equal to the protective target.
- There is some available discretionary storage.

If an address space is already going through an OK1 action, the action continues for that space. Otherwise the following criteria are used to pick new address space to go through an OK1 action:

- There must be enough discretionary central storage for the address space to increase the protective target
- Address spaces that already have a protective target are chosen over those that do not
- Address spaces that do not have a restrictive target are chosen over those that do

If no address space meets these criteria, no OK1 action is done.

The OK1 action is ended when one of the following is true:

- An OK1 point is plotted on the address spaces central paging plot
- The address space is chosen to be "squeezed" because of too much system unmanaged paging
- UIC steal could not free enough frames

2.4.8 System is Under-Utilized

The next condition verified is if the system is being under-utilized. In the affirmative case, more work should be brought into the system by raising a period's MPL. The system is considered under-utilized if all of the following conditions are true:

- On average during the last RM2 interval no more than a few address spaces were waiting for the CPU.
- The system unmanaged paging cost was under 10%.
- The amount of central storage fixed is below the RCT low thresholds (RCVFXIP, RCVMFXA).

- The system is not and has not recently been in a storage shortage.
- The current point on the system paging responsiveness plot is below the add work point.

If all these conditions are met `select_receiver` is called to find, in receiver order, a period with at least one out-and-ready user. If such a period is found, its MPL out target is raised by one.

2.4.9 Phase Change

The next action is to attempt a “phase change.” The idea behind phase change is for an address space that has had a restrictive target for a long time, to increase its central and/or processor storage restrictive target by a delta value. This is done to detect if the address space’s paging characteristics have changed such that it would run faster with just a few more frames. Without increasing the restrictive target we might never find this out since the restrictive target would stop it from getting these frames. If an address space has never gone through a phase change, it is eligible 3 minutes after getting a restrictive target. It is not eligible again for 6 minutes, then 9 minutes and so on.

Jobs are selected for a phase change by the following criteria:

- They have had a restrictive target long enough
- There is enough discretionary storage to raise the restrictive target in question (central or processor storage)

2.4.10 Working Set Management

Finally if no other actions have been taken and there are monitored address spaces, the Working Set Management (WSM) algorithm is run. WSM’s job is to reduce the amount of CPU time the system spends on paging by efficiently allocating storage among monitored address spaces. The primary functional difference between WSM in goal mode and compatibility mode is that in goal mode WSM deals with efficiently allocating expanded in addition to AUX storage.

Consider the following example showing how WSM can greatly increase the amount of productive work done by a system. On a system with 120MB of central storage, 3 jobs are run that getmain a 50MB area and reference each page in this area sequentially in a tight loop. Without WSM all three jobs might be run simultaneously with none of the jobs able to keep their storage. Each job would spend most of its time paging and almost all of the system’s CPU time will be spent paging. WSM on the other hand will understand that two of these jobs will fit without paging and keep one job swapped out. In this case most of the system’s CPU time will go towards running the application’s code of the jobs getting productive work done. Exchange swapping will ensure that each job gets a chance to run.

If WSM wants an address space to have more central storage, it raises the address space’s protective central storage target. Similarly WSM raises an address space’s protective processor storage target to give an address space more processor storage. When WSM wants to take central or processor storage from an address space, it gives the address space the appropriate restrictive target. If WSM decides an address space can do very little productive work with the storage it has been allocated, it swaps the address space out. For a batch job, WSM can keep the address space swapped out up to ten minutes past the

time it would normally be swapped in. In the case of a TSO user, the address space can only be swapped out an extra 30 seconds.

Chapter 3. Workload Manager Performance Monitoring

This chapter provides an overview of the RMF reports that has been enhanced to support the MVS Workload Manager (WLM). It also contains an introduction of the SMF type 99 records and examples of how to interpret the trace data output.

The major objectives of RMF Version 5 Release 1.0 are to support:

- The concept of a single image for performance management
- The Workload Manager

RMF provides a sysplex scope for its performance management functions. Single system performance reports are combined into sysplex-wide overview reports with the capability to select different detail levels, in terms of systems, resource and logical entities such as workloads.

The RMF data gatherers will collect the measurement data on each MVS image in the sysplex. To provide a sysplex scope for the users, the data is combined in one place accessible for the reporter sessions.

The RMF Sysplex Data Server allows access to sysplex-wide collected measurement data for any RMF reporter session. Each system individually collects and records measurement data to avoid single point of failures and provides continuous availability of performance management in the sysplex.

The RMF postprocessor will make use of the new function in order to immediately process the generated data for either sysplex-wide or single-system oriented reports. The RMF Monitor III reporter uses the functions for remote reporting of any member in the sysplex as well as to generate reports with a sysplex-wide scope.

With the introduction of MVS Workload Manager, the System Resources Manager (SRM) can be managed in two modes:

Two modes of operation

- Compatibility mode, where the system resource management is determined by the IPS and ICS PARMLIB members
- Goal mode, where the system resource management is determined by the goal-oriented service policies.

To allow migration in easy stages, RMF supports both compatibility mode and goal mode in a non-disruptive way. This means that the data gatherer function will automatically recognize the new mode and switch to the new collection mode. The reporters will change their report layout according to the new mode.

We will look briefly at the RMF monitor reports that have been enhanced to support the MVS Workload Manager.

3.1 RMF Reports in Compatibility Mode

When running in compatibility mode, there are no changes to the RMF Monitor II reports. In RMF Monitor I/Postprocessor there are some new fields in the Workload Activity report.

3.1.1 Workload Activity Report-Compat Mode

- The new field **MODE=** indicates the mode the system was running in at the time the data was gathered.
- The new field **EX VEL%** represents the execution velocity for a performance group.
- The new field **SRVCLASS=** represents the service class to which the subsystem (CICS or IMS) transaction are classified.

The example in Figure 18 shows a report with the new fields.

WORKLOAD ACTIVITY														PAGE 4	
MVS/ESA	SYSTEM ID UNKN			DATE 05/21/94	INTERVAL 04.59.985	MODE=COMPAT									
SP5.1.0	RPT VERSION 5.1.0			TIME 10.30.09	IPS = IEAIP5B										
OPT = IEAOPTGM	REPORT BY PERFORMANCE GROUP			SERVICE DEFINITION COEFFICIENTS				SU/SEC=2656.0							
ICS = IEAICS51	PERIOD			IOC = 5.0 CPU = 10.0 SRB = 10.0				MSO = 3.0000							
PGN	PGP	DMN	TIME SLICE GROUP	INTERVAL	SERVICE	AVERAGE ABSORPTION, AVG TRX SERV RATE, TCB+SRB SECONDS, %		PAGE-IN RATES		STORAGE		TRANSACTIONS		AVG TRANS. TIME, STD. DEVIATION HHH.MM.SS.TTT	
0081	1	081	**	IOC	0	ABSRPTN	82	SINGLE	0.84	AVERAGE	60.39	AVG	0.00	TRX 000.00.00.000	
				CPU	470	TRX SERV	82	BLOCK	0.00			MPL	0.00	SD 000.00.00.000	
				MSO	98	TCB	0.0	HSP	0.00	TOTAL	0.35	ENDED	0		
				SRB	13	SRB	0.0	HSP MISS	0.00	CENTRAL	0.22	END/SEC	0.00	QUE 000.00.00.000	
				TOT	581	TCB+SRB%	0.0	EXP SNGL	1.98	EXPAND	0.13	#SWAPS	3		
				PER SEC	0	EX VEL%	0.0	EXP BLK	0.00					TOT 000.00.00.000	
				ZEROS											
0082	ALL	ALL	ALL	THRU 0099											
0100	1	007	**	IOC	325.3K	ABSRPTN	77,886	SINGLE	0.00	AVERAGE	7,288.4	AVG	2.00	TRX 000.00.00.000	
				CPU	1741K	TRX SERV	77,886	BLOCK	0.00			MPL	2.00	SD 000.00.00.000	
				MSO	44561K	TCB	60.7	HSP	0.00	TOTAL	14,584	ENDED	0		
				SRB	124.9K	SRB	4.3	HSP MISS	0.00	CENTRAL	8,251.7	END/SEC	0.0	QUE 000.00.00.000	
				TOT	46752K	TCB+SRB%	3.0	EXP SNGL	2.95	EXPAND	6,332.1	#SWAPS	0		
				PER SEC	155,847	EX VEL%	42.5	EXP BLK	0.00					TOT	
000.00.00.000	0101	ALL	ALL	ALL	THRU 0199										
				ZEROS											
SUBSYS = CICS	TRXCLASS =		ACCTINFO = NO												
USERID =	TRXNAME =		SRVCLASS = CICSRTX												
0200	1	**	**	IOC	0	ABSRPTN	0	SINGLE	0.00	AVERAGE	0.00	AVG	0.00	TRX 000.00.00.094	
				CPU	0	TRX SERV	0	BLOCK	0.00			MPL	0.00	SD 000.00.00.068	
				MSO	0	TCB	0	HSP	0.00	TOTAL	0.00	ENDED	14,126		
				SRB	0	SRB	0.0	HSP MISS	0.00	CENTRAL	0.00	END/SEC	47.08	QUE 000.00.00.000	
				TOT	0	TCB+SRB%	0.0	EXP SNGL	0.00	EXPAND	0.00	#SWAPS	0		
				PER SEC	0	EX VEL%	0.0	EXP BLK	0.00					TOT 000.00.00.182	
				ZEROS											
0201	ALL	ALL	ALL	THRU 0299											
				ZEROS											

Figure 18. Workload Activity Report in Compatibility Mode

The **EX VEL%** and **SRVCLASS** may be used for guidance in setting goal mode objectives.

In this example, two CICS AORs are running in performance group 100 and RMF provides its usual service and storage data for address spaces. The new *execution velocity* data shows that over this interval, the two spaces have run

with a velocity of 42.5. RMF also shows information about some of the transactions running in those regions in Report PGN 200.

The *SRVCLASS* parameter lets you associate a service class with a report performance group in compatibility mode. In this example we defined two service classes for IMS work in our service definition, and the following in the IEAICSxx member:

```

EXAMPLE
SUBSYS=IMS
SRVCLASS=IMSHI,RPGN=100
SRVCLASS=IMSLow,RPGN=200
  
```

3.2 RMF Reports in Goal Mode

When running in goal mode, there will be no new RMF Monitor II reports; however there will be some minor changes to existing reports:

- ASD, ASDJ, ASRM, ASRMJ reports will replace DMN and PGN by *SRVCLASS*
- DDMN and TRX reports are not available

The example in Figure 19 shows the ASRM report with the new field *SRVCLASS*.

RMF MONITOR II HARDCOPY LOG												
MVS/ESA		SYSTEM ID UNKN				SESSION NAME AB						
SP5.1.0		RPT VERSION 5.1.0				DATE 05/19/94						
						TIME 14.43.47						
+++++++										+++++++ TOTAL MODE	+++++++ ASRM	+++++++
+++++++										+++++++ 14.43.47	+++++++	+++++++
14:43:47	S	TRANS	TRANS	TX	TX	TX	TX	TX	TX	SESS		
JOBNAME	<u>SRVCLASS</u>	P	ACTIVE	CUR	RES	CT	SC	CPU	MSO	IO	SR	TOTA
MASTER	SYSTEM	1	00:11	00:11:32	3	0	19623	32512	460	127124	8.808M	
PCAUTH	SYSSTC	1	00:11	00:11:32	3	0	0	0	0	0	0	107
RASP	SYSTEM	1	00:11	00:11:32	3	0	0	0	0	550	6845	
TRACE	SYSSTC	1	00:11	00:11:32	3	0	0	0	0	0	0	47
DUMPSRV	SYSTEM	1	00:11	00:11:32	3	0	0	0	0	0	0	2962
XCFAS	SYSTEM	1	00:11	00:11:32	3	0	7632	20923	0	543	612237	
GRS	SYSTEM	1	00:11	00:11:32	3	0	0	0	0	10	1063	
SMXC	SYSTEM	1	00:11	00:11:32	3	0	2742	901	0	1036	146713	
SYSBMAS	SYSSTC	1	00:11	00:11:32	3	0	131	40	0	0	92547	
CONSOLE	SYSTEM	1	00:11	00:11:32	3	0	10591	7100	885	795	1.344M	
WLM	SYSTEM	1	00:11	00:11:32	3	0	227918	266286	0	3721	991487	
ANTMAIN	SYSTEM	1	00:11	00:11:32	3	0	0	0	0	6	301	
ALLOCCAS	SYSSTC	1	00:11	00:11:32	3	0	0	0	0	0	7292	
IOSAS	SYSTEM	1	00:11	00:11:32	3	0	0	0	0	0	24186	
VTAMR	VTAMDB2	1	00:11	00:11:32	3	0	7574	9520	0	8243	8.448M	
SMF	SYSTEM	1	00:11	00:11:32	3	0	462	293	30	2072	157971	
LLA	SYSSTC	1	00:11	00:11:32	3	0	1384	3610	500	141	546332	
VLF	SYSSTC	1	00:11	00:11:32	3	0	336	9058	0	0	214548	
JES2	SYSSTC	1	00:11	00:11:32	3	0	72822	269980	17140	5136	50.01M	
RACF	SYSSTC	1	00:11	00:11:32	3	0	0	0	0	0	529	
CATALOG	SYSTEM	1	00:11	00:11:32	3	0	142022	732397	3940	4197	42.89M	
M1A2	BATCHLOW	1	00:11	00:11:32	3	0	3.532M	195.0M	1.192M	346836	1284M	
M1T1	BATCHLOW	1	00:11	00:11:32	3	0	1.851M	35.60M	0	1.033M	297.5M	
M1A1	BATCHLOW	1	00:11	00:11:32	3	0	3.095M	59.36M	61282	87150	450.1M	
M1F1	BATCHLOW	1	00:11	00:11:32	3	0	2.524M	117.7M	1.132M	387441	782.0M	
U3C0004	TSO	1	00:00	00:00:00	228	0	19	54	0	0	424442	
XA5MSTR	VTAMDB2	1	00:11	00:11:32	3	0	1643	915	0	935	300968	
IRLMWFI	VEL50	1	00:11	00:11:32	3	0	0	0	0	217	3397	

Figure 19. ASRM Report in Goal Mode

In the RMF Monitor I postprocessor, the Workload Activity report is different from the compatibility version, and exists only as a postprocessor report. This report provides data/numbers documenting the response time, service rate, transaction rate, and so forth, that particular workloads are achieving. Furthermore the

report compares the performance goals, as defined in the service policy, with actual values achieved by all work in particular class periods.

The general appearance of the report is shown in Figure 20 on page 63.

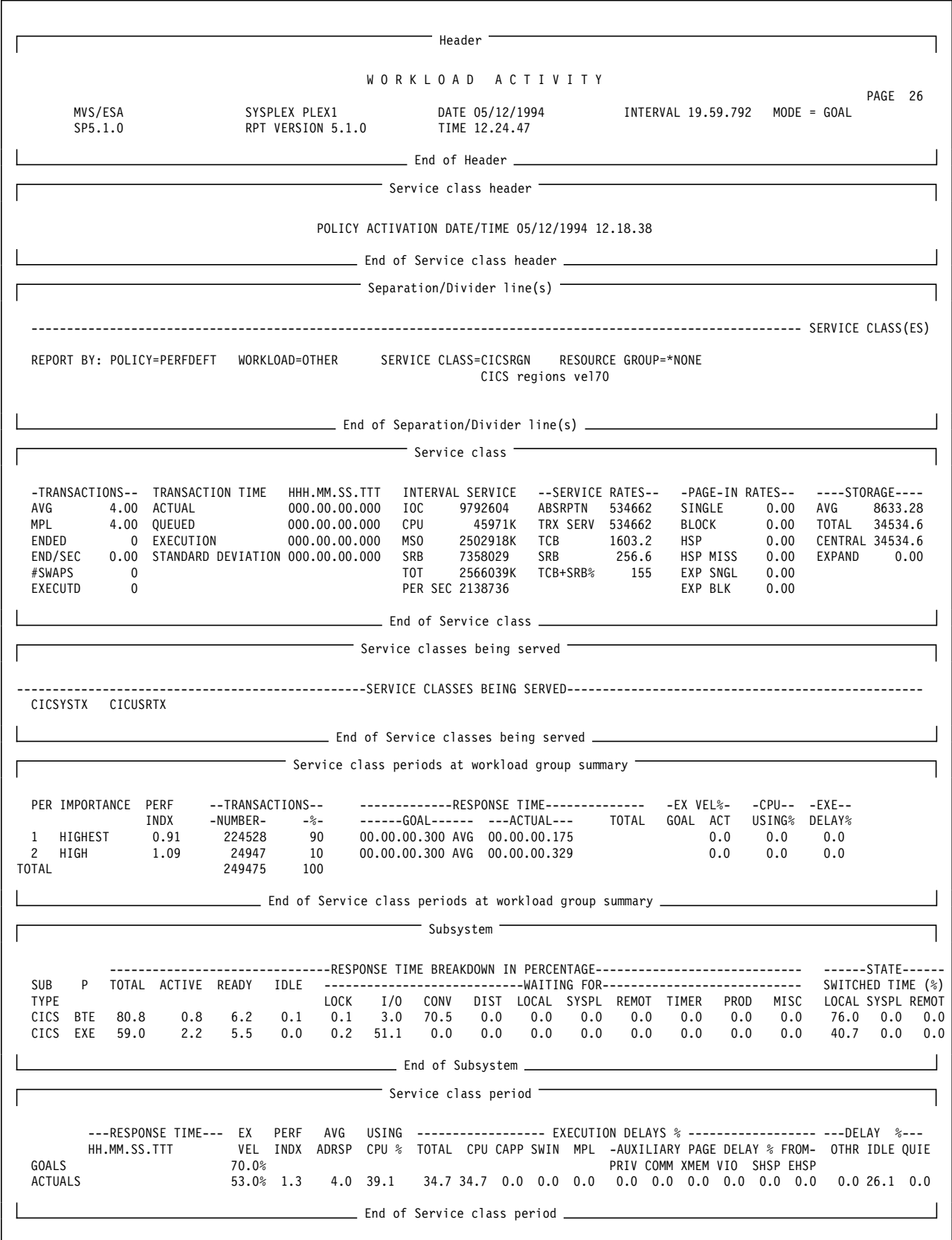


Figure 20 (Part 1 of 2). Workload Activity Report in Goal Mode

RCLASS All report classes defined in a service policy

3.2.1 RMF Monitor III Reports in Goal Mode

All Monitor III reports are available in goal mode. The reports keep their single system view, but the report layout may differ. The arrangement of reports by domain and performance group is replaced by workloads, service classes and service class periods. Depending on the data presented in the report, changes are made to add useful goal mode information.

Three new reports are available for goal mode and a new Data Index screen, all of which have a sysplex-wide reporting scope:

- Sysplex Performance Summary report, shown in Figure 21
- Response Time Distribution report, shown in Figure 22 on page 66
- Work Manager Delays report, shown in Figure 23 on page 67
- Sysplex Data Index screen

Figure 21 shows the Sysplex Performance Summary report. This report provides an overview of workload groups, service classes, service class periods and report classes. It includes a goal versus actual comparison for each service class period. You can compare different goals by using the performance index (PI).

Furthermore, the response time for all groups is calculated independently of any specified goals, and a transaction rate is provided to enable you to weigh the importance of the statistics shown.

```

RMF 5.1.0 Sysplex Summary - WTSCPLX1          Line 1 of 3
Command ==>                               Scroll ==> CSR

WLM Samples: 400   Systems: 1   Date: 06/29/94   Time: 14.06.40   Range: 100   Sec

>>>>>>>XXXXXXXXXXXXXXXXXXXX<<<<<<<<

Service Definition: CICSPOL                Installed at: 06/24/94, 11.57.59
Active Policy: CICMAJ                      Activated at: 06/24/94, 11.58.41

----- Goals versus Actuals -----   Trans --Avg. Resp. Time-
Exec Vel  --- Response Time --- Perf  Ended  Queue Activ Total
Name      T I  Goal Act  ---Goal--- --Actual--  Indx  Rate   Time  Time  Time
TSO       W   73
TSO       S   80
          1 3   50 0.100 AVG 0.258 AVG 2.58 0.400 0.087 0.170 0.170
          2 3   100 1.000 AVG 2.529 AVG 2.53 0.010 0.000 2.520 2.520
          3 4   100 3.000 AVG 9.588 AVG 3.20 0.050 0.000 9.580 9.580
TSOSPECL S 1   99 60                1.65 0.550 0.000 0.341 0.341
CATALOG  R   N/A 100  N/A                0.000 0.000 0.001 0.001
CICSFRPT R   N/A 100  N/A                0.010 3.516 2.612 2.612
CICSRDFT R   N/A 0    N/A                0.100 0.000 0.003 0.003
CLASSA   R   N/A 100  N/A                0.010 3.518 8.346 8.346

```

Figure 21. Sysplex Performance Summary Report

Figure 22 on page 66 shows the Response Time Distribution report. This report enables you to analyze the distribution of response time to see whether a response time goal was met and, if not, how close it came to meeting the goal. This response can also be used to fine-tune response time goals.

Two levels of detail are shown:

- A graph shows the distribution of response time for all systems in a sysplex which have data available in the service class period.
- The details of how each system contributed to the overall response time.

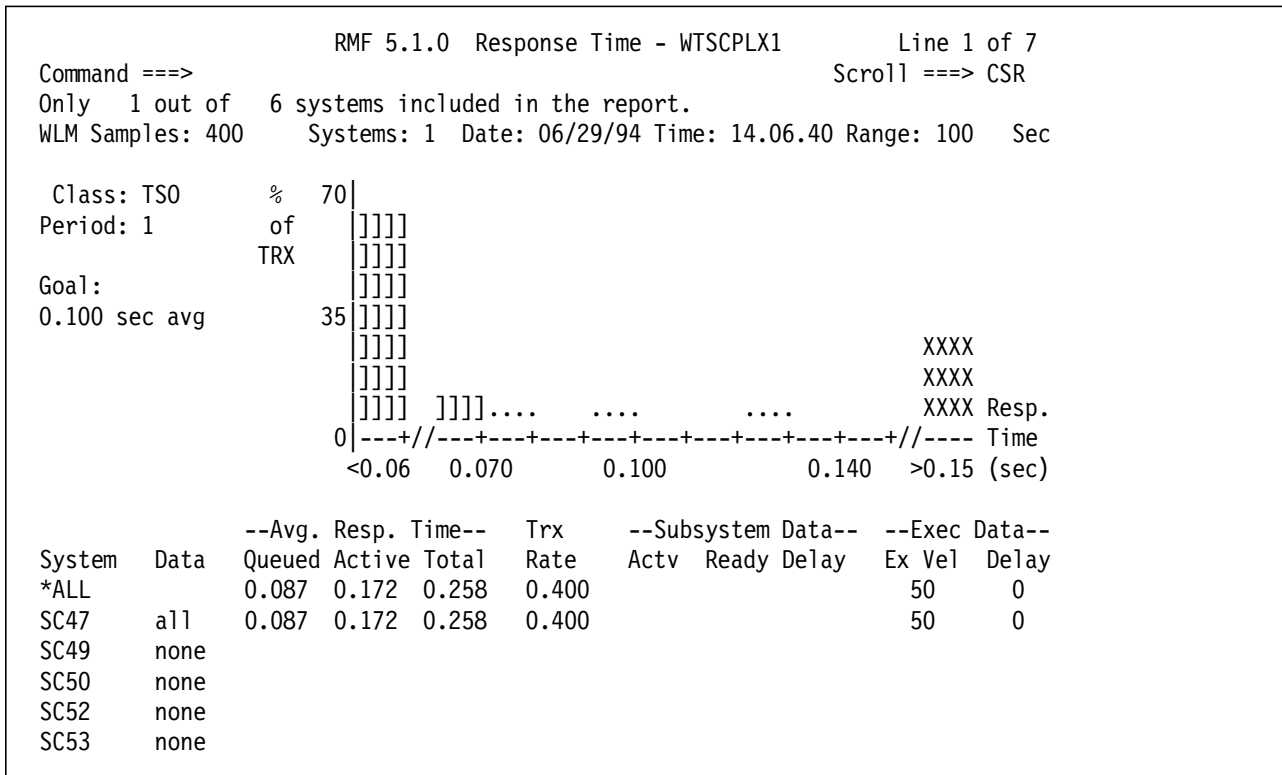


Figure 22. Response time Distribution Report

Figure 23 on page 67 shows the Workload Manager Delays report. This report shows details for resource-manager or work-manager oriented subsystems and is intended as a basis to start tuning. Using the real-time data, you can use this report to track problems as they happen.

The report shows the average transaction response time and how the various transaction states contribute to it. Furthermore it lists the address spaces that have been used by the transactions.

```

RMF 5.1.0 Sysplex Summary - WTSCPLX1           Line 1 of 3
Command ==>>>                               Scroll ==>>>

WLM Samples: 400   Systems: 1   Date: 06/29/94   Time: 14.06.40   Range: 1

Class:  POSMULTI   Period: 1           Avg. Resp. time: 0.587 sec for 12473 TRX.
Goal:   0.500 sec average               Avg. Exec. time: 0.379 sec for 12389 TRX.
Actual: 0.587 sec average               Abnormally ended:           0 TRX.

Sub P  -----Response time Breakdown (in %)------Switched--
Type   Tot Act Rdy Idle -----Delayed by-----Time (%)
              Lock I/O Conv Dist Sess Time Prod Misc Loc Sys Rem

CICS B   84  9   0   0   5  0  65  0  0  0  0  5  40 25  0
CICS X   43  4   8   0   6 18  0  0  0  0  1  6  0  0  0
IMS X    16  5   0   0   0 10  0  0  0  0  0  0  0  0  0

-----Address Spaces serving this Service Class Posmulti-----
Jobname   Asid  system  Serv-Class  Service  Proc-usg  Veloc  Capp  Quies

CICSTOR1  0035  MVS1    CICSTOR    36       6         35     0     0
CICSTOR2  0100  MVS2    CICSTOR    64       11        50     0     0
CICSAOR1  0405  MVS2    CICSAOR    21       8          5     18     0
CICSAOR2  0058  MVS2    CICSAOR    53       2         33     18     0
IMSDBCTL  0130  MVS2    SYSSTC     48       7         73     0     0

```

Figure 23. Work Manager Delays Report

For a complete description of the fields of all three RMF Monitors refer to *MVS/ESA Analyzing Resource Measurement Facility Version 5 Reports*

3.3 SMF Record Type 99

SMF record type 99 provides detailed audit information for work run in workload management goal mode. They can be useful in analyzing and understanding the performance characteristics of a customer’s workload. The records contain performance data for each service class period, a trace of SRM actions, the data SRM used to decide which actions to take, and the internal controls SRM uses to manage work. This can help you to determine in detail what SRM is doing to meet your work’s goals with respect to other work, and the types of delays the work is experiencing.

You should ask for SMF type 99 records only when you want this detailed information. For general reporting and tuning information for a system in goal mode, you can use SMF type 72 records.

SRM writes type 99 records in goal mode for each *policy interval*, or *approximately* once every 10 seconds.

There are four subtypes of SMF type 99 records:

- Subtype 1** Contains system level data, the trace of SRM actions, and data about resource groups. A subtype 1 record is written every policy interval.
- Subtype 2** Contains data for service classes. A subtype 2 record is written every policy interval for each service class if any period in the service class had recent activity.
- Subtype 3** Contains service class period plot data. A subtype 3 record is written every policy interval for each service class if any period in the service class had recent activity and plot data.
- Subtype 5** Contains data about monitored address spaces. A subtype 5 record is written each policy interval for each swapped-in monitored address space.

3.3.1 Identifying Work in SMF type 99 Records

The work can be identified by:

- Service class name
- Service class performance period number
- Resource group name
- Address space name

3.3.2 Identifying Server Service Classes

Some service classes are server service classes, that is they are service classes representing address spaces doing work on behalf of transactions. You can determine whether a service class represented in the subtype 2 period data is a server by first checking the goal type in the service class period section. If the goal type is 0, indicating a system or server service class, you next check whether there are any server data entries. If there are entries, then the service class period is a server, and the server data describes the service classes being served by this server.

If the goal type is 0, and there are no server entries, then the service class is a system service class.

3.3.3 Identifying Internal Service Classes

SRM groups some work into internal service classes. The internal service class names are:

- \$SRMDInn** Resource group discretionary — This service class contains all work in a resource group with a discretionary goal. There is one \$SRMDInn class per resource group with discretionary work.
- \$SRMDI00** General discretionary — This service class contains all work assigned a discretionary goal, but not assigned a resource group.
- \$SRMDISC** SYSOTHER service class — This service class contains all work not otherwise assigned to a service class.
- \$SRMSnnn** Server — This service class contains all address spaces serving the same set of service classes. That is, the server address space(s) could be serving more than one service class. For example, two AORs may be serving the same three CICS service classes. Those two AORs make up one \$SRMSnn service class. An address space can belong to one internal server service class, but can move from one class to another. The number of \$SRMSnnn service classes

depends on how many external service classes are served and by how many combinations of server address spaces there are.

\$SRMDUMP SDUMP — This service class contains only SDUMP, if it has not been classified in the service definition.

\$SRMBEST This service class contains the special system component address spaces, unless they are explicitly assigned to a service class in the service policy. This is the SYSTEM service class.

\$SRMGOOD This service class contains the STC work unless they are explicitly assigned a service class or they are one of the special SYSTEM address spaces.

3.3.4 Interpreting SMF Record Type 99 Traces

This section shows examples where we could see the controls that SRM uses to manage work. All changes to control values have trace entries in the subtype 1 record. The trace codes describe which action SRM took or is considering taking to process work.

The SRM controls are:

- Dispatching priority
- MPL targets
- Swap protect time
- Expanded storage policy
- Storage targets
- Cap slices

SRM creates the following plots to track how well work is being processed:

- System paging delay plots
- Period MPL delay plot
- Period ready user average plot
- Period swap delay plot
- Period paging rate plot
- Period proportional aggregate speed plot
- Address space paging plot

The SRM actions are:

- Policy adjustment
- Resource adjustment

Note: The SRM controls, actions and plots are explained in detail in Chapter 2, “System Resource Manager MVS/ESA Version 5” on page 9

The examples show information from SMF type 99 records that were combined, reduced and displayed in a report format with an internally developed program.

3.3.5 Action Trace Example

The following examples show how to use the trace data to understand what actions SRM is taking, why those actions are taken, and which work is affected.

Figure 24 shows the subtype 1 trace data output:

```

94.167 13:18:37 PS1
-----
GROUP   CLASS   P ADDRSP   SPI  LPI  GSR  RID CODE ACTION
$SRMS01F 1 M1A2    75  75    116 6510 hsk_sl_dec_ici_tar
IMS1     1        291 291    116 240 pa_grec_cand
$SRMS021 1        291 291    116 280 pa_rrec_cand
BATCHLOW 1         4   4    116 880 pa_pro_rdon_cand
$SRMS021 1        291 291    116 620 pa_pmuo_rec
BATCHHI  1        125 125    116 960 pa_pro_unc_sec_don
BATCHLOW 1         8   8    116 940 pa_pro_unc_don
TSO      2         49 49    116 960 pa_pro_unc_sec_don
TSO      3         97 97    116 960 pa_pro_unc_sec_don
$SRMS021 1        291 291    116 750 pa_pro_incp_rec
IMS1     1        227 227    116 936 pa_pro_served_grec
IMS      1        21  21    116 936 pa_pro_served_grec
$SRMS023 1        21  21    116 960 pa_pro_unc_sec_don
IMS      1        21  21    116 933 pa_pro_served_gdon

```

Figure 24. Subtype 1 Trace Data Output

The data has the following headings:

- GROUP** Resource group name.
- CLASS** The service class name.
- P** The service period number within the service class.
- ADDRSP** Address space name.
- SPI** The sysplex performance index for the service class period traced.
- GSR** Group service resource.
- RID** Resource interval identifier.
- LPI** The local performance index for the service class period traced.
- ACTION** The action code and mnemonic.

Figure 25 shows the subtype 2 service class period output.

```

PERIOD SUMMARY
CLASS  P  I  N  SC  SPI  LPI  DP  SERV  MDP  MPLI  MPLO  SWPT  PSI  EXPPOL  DELAY SAMPLES/SERVES/SERVERS
BATCHHI 1 3 Y -99 50 50 243 6544 252 4 999 0 L 0/7/0 7/0 7/0 CPUD/588 OTHR/299
BATCHLOW 1 3 Y -99 5 5 243 2101 74 1 10 0 L 0/1/0 1/0 1/0 CPUD/115 OTHR/43
CICSRGN 1 2 N -99 318 318 247 0 1 2 999 0 L 0/0/0 0/0 0/0 CPUD/311
CICSYSTX 1 2 Y -99 40 40 $SRMS020/134
CICUSRXTX 1 2 Y -78 70 70 $SRMS020/138 $SRMS01F/1658
IMS 1 2 Y -99 1 1 $SRMS023/8108 $SRMS021/*****
IMSCTL 1 2 N -99 250 250 254 0 1 1 999 0 L 0/0/0 0/0 0/0 CPUD/76
IMSMP 1 2 N -99 135 135 247 0 1 28 999 0 L 0/0/0 0/0 0/0 OTHR/11910 CPUD/129
IMS1 1 2 Y -99 45 45 $SRMS022/621 $SRMS021/14688
TSO 1 2 Y -99 71 71 245 5385 106 2 999 2440 0 L L L L CPUD/149 OTHR/106 ASWP/6
TSO 2 2 Y -99 36 36 243 1674 53 1 999 0 0 L L L L OTHR/885 CPUD/476
TSO 3 2 Y -99 59 59 243 3939 118 2 999 0 0 L L L L CPUD/521 OTHR/506

```

Figure 25. Subtype 2 Trace data output

The subtype 2 service class policy period output has the following headings:

CLASS	The service class name.
P	The service period number within the service class.
I	The service class period's importance, with 1 being highest.
N	The "needs help" indicator. Y means the service class period can be helped and N means that the period cannot be helped.
SC	The skip clock— reduced by 1 every policy adjustment cycle. No policy adjustment action is taken until the clock is zero or less.
SPI	The sysplex performance index (scaled by 100) for the service class period traced.
LPI	The local performance index (scaled by 100) for the service class period traced.
DP	Dispatching priority
SERV	Service units consumed in this interval.
MDP	Maximum demand percentage of CPU time.
MPLI	MPL-in target.
MPLO	MPL-out target.
SWPT	Swap protect time in milliseconds.
PSI	Protective processor storage target.
EXP POL	The expanded storage access policies, in the following order from left to right: <ul style="list-style-type: none">• Swap working set pages• Stolen pages and swap trim (demand pages)• VIO pages• Hiperspace pages Where P=protected, L=LRU, and S=space available. If there are policies for individual address spaces, then there is a triplet indicating the number of address spaces with each setting (P,L,or S).
DELAYS SAMPLES/SERVES/SERVERS	The count of delay samples with either the reasons for the delays or the names of the served classes.

3.3.6 Policy Adjustment/Dispatching Priority Example

This example shows the dispatching priority control SRM uses when it finds that the bottleneck is caused by processor delay. There are more examples in the chapter "Using SMF Record 99" in the manual *Workload Management Services*.

Figure 26 on page 72 shows the first DP Period Summary.

PERIOD SUMMARY													DELAY SAMPLES/SERVES/SERVERS					
CLASS	P	I	N	SC	SPI	LPI	DP	SERV	MDP	MPLI	MPLD	SWPT	PSI	EXPPOL				
BATCHHI	1	3	N	-47	5	5	243	0	1	0	999	0		L 0/0/0 0/0 0/0			OTHR/2	
BATCHLOW	1	3	N	-47	2	2	243	0	1	0	10	0		L 0/0/0 0/0 0/0			OTHR/38 CPUD/6	
CICSRGN	1	2	N	-41	318	318	247	0	1	2	999	0		L 0/0/0 0/0 0/0			CPUD/311	
CICSYSTX	1	2	Y	-37	40	40												\$\$SRMS020/67
CICUSRTX	1	2	Y	0	144	144												\$\$SRMS020/111 \$\$SRMS01F/1449
IMS	1	2	Y	-47	2	2												\$\$SRMS023/4275 \$\$SRMS021/79029
IMSCTL	1	2	N	-43	250	250	249	0	1	1	999	0		L 0/0/0 0/0 0/0			CPUD/76	
IMSMP	1	2	N	-37	135	135	247	0	1	28	999	0		L 0/0/0 0/0 0/0			OTHR/11910 CPUD/129	
IMS1	1	2	Y	-47	62	62												\$\$SRMS022/388 \$\$SRMS021/8136
TSO	1	2	Y	-47	67	67	245	5598	134	2	999	2440	0	L L L L L			OTHR/85 CPUD/71 MPLD/1	
TSO	2	2	Y	-47	34	34	243	1788	41	1	999	0	0	L L L L L			OTHR/501 CPUD/264	
TSO	3	2	Y	-47	29	29	243	3555	95	1	999	0	0	L L L L L			OTHR/473 CPUD/270	
\$\$SRMS01F	1	2	Y	-42	144	144	247	8762	192	999	999	0		L 3/0/0 3/0 3/0			CPUD/110	
																		CICUSRTX/1449
\$\$SRMS020	1	2	Y	-42	144	144	251	2589	58	999	999	0		L 1/0/0 1/0 1/0			CPUD/52	
																		CICUSRTX/111 CICSYSTX/67
\$\$SRMS021	1	2	Y	-42	62	62	241	3523	92	999	999	0		L 3/0/0 3/0 3/0			CPUD/329 OTHR/124	
																		IMS1/8136 IMS/79029
\$\$SRMS022	1	2	N	-42	62	62	254	135	4	999	999	0		L 4/0/0 4/0 4/0			OTHR/1150	
																		IMS1/388
\$\$SRMS023	1	2	Y	-42	2	2	241	7538	162	999	999	0		L 51/0/0 51/0 51/0			OTHR/5737 CPUD/554	
																		IMS/4275

Figure 26. DP Period Summary First Example

Figure 27 shows the trace output of the SRM actions.

GROUP	CLASS	P	ADDRSP	SPI	LPI	GSR	RID	CODE	ACTION
94.167	13:04:58	PS1	-----						
TSO	1		67	67		227	3610	rv_hsk_inc_mpl	
CICUSRTX	1		144	144		227	240	pa_grec_cand	
\$\$SRMS01F	1		144	144		227	280	pa_rrec_cand	
TSO	1		67	67		227	880	pa_pro_rdon_cand	
\$\$SRMS01F	1		144	144		227	620	pa_pmuo_rec	
CICSRGN	1		318	318		227	960	pa_pro_unc_sec_don	
IMSMP	1		135	135		227	960	pa_pro_unc_sec_don	
TSO	1		77	77		227	940	pa_pro_unc_don	
\$\$SRMS01F	1		144	144		227	750	pa_pro_incp_rec	
CICUSRTX	1		124	124		227	936	pa_pro_served_grec	

Figure 27. DP Trace data output

Based on the information in Figure 26 and Figure 27 SRM took the following actions:

- *rv_hsk_inc_mpl* indicates a reverse housekeeping operation, incrementing the MPL for TSO users.
- *pa_grec_cand* indicates that policy adjustment has selected CICUSRTX as the goal receiver candidate. This is the service class that has the highest PI that can be helped.
- *pa_rrec_cand* indicates that \$\$SRMS01F is the resource receiver candidate. This is the DISC which represents the transactions running in the CICUSRTX external service class.
- *pa_pro_rdon_cand* indicates that TSO has been selected as the processor resource donor candidate.
- *pa_pmuo_rec* is a policy adjustment operation, assessing moving the primary processor receiver up to an occupied priority.

- *pa_pro_unc_sec_don*, unchanged secondary donor, indicates that this change is not projected to change the PI of the secondary donors, CICSRRGN and IMSMP.
- *pa_pro_unc_don*, unchanged donor, indicates that the primary donor TSO, will not be affected by this operation.
- *pa_pro_incp_rec*, increase priority for receiver, indicates that the change in dispatching priority for the address spaces in \$SRMS01F is being done.
- *pa_pro_served_grec* indicates that CICUSRTX is the goal receiver of the change in dispatching priority.

Figure 28 shows the period summary *after* the SRM actions.

PERIOD SUMMARY													DELAY SAMPLES/SERVES/SERVERS				
CLASS	P	I	N	SC	SPI	LPI	DP	SERV	MDP	MPLI	MPLO	SWPT	PSI	EXPPOL			
BATCHHI	1	3	N	-48	5	5	243	0	1	0	999	0		L 0/0/0	0/0	0/0	OTHR/2
BATCHLOW	1	3	N	-48	2	2	243	0	1	0	10	0		L 0/0/0	0/0	0/0	OTHR/38 CPUD/6
CICSRRGN	1	2	N	-42	318	318	247	0	1	2	999	0		L 0/0/0	0/0	0/0	CPUD/311
CICSYSTX	1	2	Y	-38	26	26											\$SRMS020/71
CICUSRTX	1	2	Y	0	97	97											\$SRMS020/113 \$SRMS01F/1483
IMS	1	2	Y	-48	1	1											\$SRMS023/4368 \$SRMS021/80778
IMSCTL	1	2	N	-44	250	250	254	0	1	1	999	0		L 0/0/0	0/0	0/0	CPUD/76
IMSMP	1	2	N	-38	135	135	247	0	1	28	999	0		L 0/0/0	0/0	0/0	OTHR/11910 CPUD/129
IMS1	1	2	Y	-48	46	46											\$SRMS022/396 \$SRMS021/8346
TSO	1	2	Y	-48	58	58	245	5641	123	2	999	2440	0	L L L L			OTHR/117 CPUD/97 ASWP/2
TSO	2	2	Y	-48	34	34	243	1858	42	1	999	0	0	L L L L			OTHR/521 CPUD/246
TSO	3	2	Y	-48	28	28	243	3275	91	1	999	0	0	L L L L			OTHR/493 CPUD/266
\$SRMS01F	1	2	Y	-43	97	97	247	8881	192	999	999	0		L 3/0/0	3/0	3/0	CPUD/92
																	CICUSRTX/1483
\$SRMS020	1	2	Y	-43	97	97	247	2596	59	999	999	0		L 1/0/0	1/0	1/0	CPUD/56
																	CICUSRTX/113 CICSYSTX/71
\$SRMS021	1	2	Y	-43	46	46	241	3520	93	999	999	0		L 3/0/0	3/0	3/0	CPUD/306 OTHR/164
																	IMS1/8346 IMS/80778
\$SRMS022	1	2	N	-43	46	46	254	133	10	999	999	0		L 4/0/0	4/0	4/0	OTHR/651
																	IMS1/396
\$SRMS023	1	2	Y	-43	1	1	241	7538	161	999	999	0		L 51/0/0	51/0	51/0	OTHR/7810 CPUD/516

Figure 28. DP Period Summary Second Example

3.3.7 Resource Adjustment/Working Set Management Example

This example contains OK1 and Working Set Management actions.

Resource adjustment algorithms get executed when the system detects a problem. In this case the resource adjustment routines noticed that the system paging cost was too high, and adjusted the storage targets for the address spaces.

Figure 29 on page 74 shows the period summary of the service classes active in the system.

PERIOD SUMMARY													DELAY SAMPLES/SERVES/SERVERS				
CLASS	P	I	N	SC	SPI	LPI	DP	SERV	MDP	MPLI	MPLD	SWPT	PSI	EXPPOL			
BATCHHI	1	3	N	-58	25	25	241	0	1	0	10	0	L	0/0/0	0/0	0/0	CPUD/3 OTHR/2
BATCHLOW	1	3	N	-12	200	200	247	0	1	1	999	0	L	0/0/0	0/0	0/0	OTHR/41 CPUD/6
CICSRGN	1	2	N	-54	269	269	254	0	1	2	999	0	L	0/0/0	0/0	0/0	CPUD/273 OTHR/1
CICSYSTX	1	2	Y	-58	86	86											\$SRMS020/119
CICUSRTX	1	2	Y	0	167	167											\$SRMS020/248 \$SRMS01F/1784
IMS	1	2	Y	-58	34	34											\$SRMS023/4597 \$SRMS021/97341
IMSCTL	1	2	N	-49	500	500	254	0	1	1	999	0	L	0/0/0	0/0	0/0	CPUD/137
IMSMPL	1	2	N	-46	227	227	254	0	1	28	999	0	L	0/0/0	65482/0	65482/0	OTHR/13462 CPUD/307
IMS1	1	2	Y	-58	105	105											\$SRMS022/348 \$SRMS021/10080
TSO	1	2	Y	-3	92	92	249	6032	119	2	7	10427	0	L	L	L	OTHR/93 CPUD/67 MPLD/43
TSO	2	2	Y	-23	110	110	247	1933	65	1	999	0	0	L	L	L	OTHR/616 CPUD/279
TSO	3	2	Y	-3	87	87	243	4308	149	2	999	0	0	L	L	L	CPUD/649 OTHR/361
\$SRMS01F	1	2	Y	-53	167	167	251	10304	196	999	999	0	L	3/0/0	3/0	3/0	CPUD/77
\$SRMS020	1	2	Y	-53	167	167	251	2631	60	999	999	0	L	1/0/0	1/0	1/0	CICUSRTX/1784
\$SRMS021	1	2	Y	-53	105	105	245	3415	74	999	999	0	L	3/0/0	2/0	2/0	CPUD/135
\$SRMS022	1	2	N	-53	105	105	254	140	8	999	999	0	L	4/0/0	4/0	4/0	CICUSRTX/248 CICSYSTX/119
\$SRMS023	1	2	Y	-53	34	34	241	7448	153	999	999	0	L	51/0/0	51/0	51/0	OTHR/155 CPUD/118
																	IMS1/10080 IMS/97341
																	OTHR/618
																	IMS1/348
																	OTHR/6377 CPUD/1549
																	IMS/4597

Figure 29. WSM Period Summary

Figure 30 shows the resource adjustment action trace that was recorded in the same interval.

94.172 13:01:23 PS1												
GROUP	CLASS	P	ADDRSP	SPI	LPI	GSR	RID	CODE	ACTION			
	\$SRMS021	1	IMSWLM	105	105		201	170	ra_mon_pag_cost_hi			
	\$SRMS01F	1	M1A2	167	167		202	170	ra_mon_pag_cost_hi			
	\$SRMS01F	1	M1F1	167	167		202	170	ra_mon_pag_cost_hi			
	\$SRMS021	1	IMSWLM	105	105		202	6530	ok1_inc_ici_tar			
	\$SRMS021	1	IMSWLM	105	105		202	7660	wsm_strt_ok1			
	\$SRMS01F	1	M1A2	84	84		204	4620	rv_hsk_inc_isi_tar			
	\$SRMS021	1	IMSWLM	117	117		204	4620	rv_hsk_inc_isi_tar			
	\$SRMS021	1	DLISASH1	117	117		204	4620	rv_hsk_inc_isi_tar			
	\$SRMS023	1	MPP50WA	25	25		204	4620	rv_hsk_inc_isi_tar			

Figure 30. WSM First Trace Output

We interpret the above trace as follows:

- In Figure 29, we could see the service classes that are active, including a number of DISCs which are servers for the transaction ESCs.
- *ra_mon_pag_cost_hi* indicates that the address spaces IMSWLM, M1A2 and M1F1 have started to be monitored because their paging cost is high. Table 1 on page 75 and Table 2 on page 75 show the current values.
- *ok1_inc_ici_tar* indicates that SRM will attempt to find the central storage OK1 point of the address space IMSWLM. That is, incrementing its individual protective central storage target to the point that it is spending less than 5% of its time paging from expanded or AUX.
- *wsm_strt_ok1* shows the start of the OK1 interval.
- *rv_hsk_inc_isi_tar* indicates that SRM will increment the individual processor storage targets of the address spaces M1A2, IMSWLM, DLISASH1 and MPP50WA.

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
MIA2	6391	49	111	0
M1F1	4802	58	100	0
IMSWLM	4937	1309	151	3

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
MIA2	10889	0	112	0
M1F1	9382	0	100	0
IMSWLM	5676	0	145	0

Note that the IMSWLM address space has the highest paging cost and this is the address space that WSM will attempt to fix by adding storage until it is "OK."

Ten seconds later, we get the next trace, shown in Figure 31.

```

94.172 13:01:33 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
  $SRMS021 1 IMSWLM 117 117 206 7570 wsm_end_ok1_run_ok
  $SRMS01F 1 MIA2 91 91 209 4620 rv_hsk_inc_isi_tar
  $SRMS020 1 M1T1 91 91 209 4620 rv_hsk_inc_isi_tar
  $SRMS021 1 IMSWLM 100 100 209 6510 hsk_sl_dec_isi_tar
  $SRMS023 1 MPP50WA 16 16 209 4620 rv_hsk_inc_isi_tar
  $SRMS023 1 MPP50WA 16 16 209 4620 rv_hsk_inc_isi_tar

```

Figure 31. WSM Second Trace Output

We see the following:

- *wsm_end_ok1_run_ok* indicates the end of the OK1 interval for this address space. The paging cost has dropped below 5% of its total time.
- *rv_hsk_inc_isi_tar* shows SRM decided to increase the protective processor storage target of MIA2, M1T1 and MPP50WA. This target is similar to the minimum storage isolation value in compatibility mode and applies to both central and expanded storage.
- *hsk_sl_dec_isi_tar* indicates SRM starts to decrease in small quantities the central storage target of the address space IMSWLM. The actual values are shown in Table 3 and Table 4 on page 76.

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
IMSWLM	4958	309	112	0

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
IMSWLM	5676	0	110	0

Here we see that the address space IMSWLM has more storage (it has increased from 4937 to 4958 pages) and the paging cost has dropped from 157 ms to 112 ms.

Ten seconds later, the third trace output in Figure 32 shows the WSM actions. Let's take a look at this trace.

```

94.172 13:01:43 PS1 -----
GROUP  CLASS  P ADDRSP  SPI  LPI  GSR  RID CODE ACTION
      $SRMS01F 1 M1F1    91   91    213 6630 wsm_inc_ici_tar
      213 7710 wsm_use_disc_cent
      $SRMS020 1 M1T1    93   93    214 4620 rv_hsk_inc_isi_tar
      $SRMS023 1 MPP50WA  7    7    214 4620 rv_hsk_inc_isi_tar
      $SRMS023 1 MPP50WA  7    7    214 4620 rv_hsk_inc_isi_tar
      $SRMS023 1 MPP50WA  7    7    214 4620 rv_hsk_inc_isi_tar
      $SRMS021 1 IMSWLM  88   88    214 6510 hsk_sl_dec_ici_tar

```

Figure 32. WSM Third Output Trace

We see the following:

- *wsm_inc_ici_tar* indicates the protective central storage of M1F1 will be incremented. See Table 5 and Table 6.
- *wsm_use_disc_cent* shows SRM is using the discretionary central storage.
- *rv_hsk_inc_isi_tar* shows the increment of the individual protective processor storage target of M1T1 and MPP50WA.

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
M1F1	4866	58	86	0
IMSWLM	4965	74	102	0

Address Space	Address Space Size in Frames	Page-in Rate	Paging Cost (ms)	Captured Time (ms)
M1F1	9382	0	86	0
IMSWLM	5676	0	101	0

Here we see that the IMSWLM address space has increased in size to 4965 pages, and its paging cost has further reduced to 101 ms. This reduction in "paging overhead" is the purpose of the resource adjustment routines.

3.3.8 Resource Adjustment/System Under-Utilized Example

When the system is under-utilized, more work should be brought into the system by raising a period MPL. The system is consider under-utilized if all of the following conditions are true:

- The number of address spaces waiting for the CPU was less than twice the number of online processors during the last interval
- The system unmanaged paging cost was under 10%
- The amount of central storage fixed is below the low thresholds
- The system is not and has not been in a storage shortage

Figure 33 shows the first period summary.

PERIOD SUMMARY														DELAY SAMPLES/SERVES/SERVERS				
CLASS	P	I	N	SC	GT	SPI	LPI	DP	SERV	MDP	MPLI	MPLO	SWPT	PSI	EXPPOL			
\$\$SRMDI00	1	6	N	-99	DIS	81	81	192	0	1	0	999	0	L	0/0/0	0/0	0/0	CPUD/374 OTHR/214 AUXC/2
RTXA	1	1	Y	3	VEL	63	63	251	1531	23	1	999	0	L	0/1/0	2/0	2/0	CPUD/60 AUXP/2 OTHR/1
TSO	1	2	Y	-99	SRT	12	12	247	35622	466	22	33	2928	0	L	L	L	OTHR/503 CPUD/435 AUXP/134
TSO	2	3	Y	-99	SRT	31	31	247	12465	175	3	13	0	0	L	L	L	OTHR/447 CPUD/125 AUXP/40
TSO	3	4	Y	-99	SRT	52	52	245	25690	592	17	33	488	0	L	L	L	CPUD/1573 OTHR/458 AUXP/13
VTAM	1	1	N	-44	VEL	50	50	249	9	1	1	999	0	S	1/0/0	0/0	0/0	

Figure 33. System Under-utilized First Period Summary

Figure 34 shows the SRM output trace.

GROUP	CLASS	P	ADDRSP	SPI	LPI	GSR	RID	CODE	ACTION
	TSO	3		52	52		112	100	ra_uu_inc_mpl
	TSO	3		52	52		115	100	ra_uu_inc_mpl

Figure 34. System Under-Utilized Trace Output

We interpret the above trace as follows:

- *ra_uu_inc_mpl* indicates SRM decided to increase the MPL of TSO period 3. This TSO period was the first service class period on the receiver list that had “out ready” users.

PERIOD SUMMARY														DELAY SAMPLES/SERVES/SERVERS				
CLASS	P	I	N	SC	GT	SPI	LPI	DP	SERV	MDP	MPLI	MPLO	SWPT	PSI	EXPPOL			
RTXA	1	1	Y	2	VEL	52	52	251	1642	24	1	999	0	L	0/1/0	2/0	2/0	CPUD/58 OTHR/1 AUXP/1
\$\$SRMDI00	1	6	N	-99	DIS	81	81	192	0	1	0	999	0	L	0/0/0	0/0	0/0	CPUD/374 OTHR/214 AUXC/2
TSO	1	2	Y	-99	SRT	23	23	247	35730	466	22	33	2928	0	L	L	L	MPLD/820 CPU/723 OTHR/619
TSO	2	3	Y	-99	SRT	46	46	247	13153	178	3	13	0	0	L	L	L	OTHR/237 CPUD/141 AUXP/36
TSO	3	4	Y	0	SRT	100	100	245	24243	562	37	55	976	0	L	L	L	CPUD/2975 MPLD1526 OTHR/535
VTAM	1	1	Y	-45	VEL	100	100	249	22	1	1	999	0	L	1/0/0	0/0	0/0	CPUD/2

Figure 35. System Under-Utilized Second Period Summary

Figure 35 shows the second period summary. In this summary we could see the MPL of TSO period 3 has been increased from 17 to 37 in the MPLI and from 33 to 55 in MPLO.

3.3.9 Policy Adjustment/Cap Slices Example

In this example we see how SRM manages resource groups defined with a maximum limit. In Figure 36 on page 79 we have a summary of the resource group data. It shows the following fields:

Group	The name of the resource group.
MIN	Minimum service rate for the resource group in unweighted CPU service units per second.
MAX	Maximum service rate for the resource group in unweighted CPU service units.
ACTUAL	Service rate received in the last policy adjustment interval in unweighted CPU service units per second.
SLICES	The number of cap slices in which work in this resource group was capped.

In this example the resource group is TSO1000 and was defined with no minimum value and with a maximum value of 1000.

```

94.166 01:47:52 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1079 . . . . .43
94.166 01:48:02 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1045 . . . . .44
94.166 01:48:12 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 985 . . . . .43
94.166 01:48:33 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1013 . . 44
94.166 01:49:13 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 961 43
94.166 01:50:25 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 883 41
94.166 01:50:56 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 897 40
94.166 01:53:09 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1015 . . . . .40
94.166 01:53:19 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1059 . . . . .41
94.166 01:53:29 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1076 . . . . .42
94.166 01:53:50 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 1079 . . . . .43
94.166 01:54:31 PS1 -----
RESOURCE GROUP DATA
GROUP      MIN      MAX  ACTUAL  SLICES
TSO1000    0      1000 939 . . . . .42

```

Figure 36. Resource Group Data

Figure 37 on page 80 shows the cap slices codes of the SRM actions to manage the resource group TSO1000.

- From 01:47:52 to 01:48:02 this resource group received a rate of **1079** service units per second and the number of cap slices was **43**. The action *pa_inc_incs* shows SRM decided to increase the cap slices of the group from 43 to 44, to reduce the service rate this group is receiving.

- From 01:49:13 to 01:50:56, *pa_cap_dec* indicates SRM decreased the cap slices from 44 to 40, permitting the group to receive a higher rate of CPU service units, because the rate was below the maximum.

```

94.166 01:48:02 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 19 8020 pa_cap_incs
94.166 01:48:12 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 24 8010 pa_cap_decs
94.166 01:48:33 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 34 8020 pa_cap_incs
94.166 01:49:13 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 54 8010 pa_cap_decs
94.166 01:50:25 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 88 8010 pa_cap_decs
94.166 01:50:56 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 103 8010 pa_cap_decs
94.166 01:53:09 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 168 8020 pa_cap_incs
94.166 01:53:19 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 173 8020 pa_cap_incs
94.166 01:53:29 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 178 8020 pa_cap_incs
94.166 01:53:50 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 188 8020 pa_cap_incs
94.166 01:54:31 PS1 -----
GROUP CLASS P ADDRSP SPI LPI GSR RID CODE ACTION
TS01000 0 208 8010 pa_cap_decs

```

Figure 37. Capping Action Trace

Chapter 4. Workload Manager Test Cases

In this chapter we will discuss the test runs performed to demonstrate how the Workload Manager (WLM) behaves in various scenarios. Several runs were performed:

- Two different, large TSO-only workloads.
- A mixed TSO, CICS, IMS and BATCH workload, with CICS defined in the policy as most important.
- A storage stressed environment where a large batch stream which significantly over-commits storage is added to a TSO workload.
- Capping workloads to investigate the new capping function in WLM.

The following MVS and subsystem versions were used:

- MVS/ESA Version 5 Release 1.0.
- CICS Version 4 Release 1.0 with VSAM.
- IMS Version 4.1 with prototype code to include WLM exploitation. DL/1 was the database manager.

In all test cases the CPU utilization was between 90% and 100%. Cached control units were used and there were no I/O bottlenecks. Also, in most runs, paging to DASD was negligible. All measurements were done on a monoplex.

Some measurements were also made in compatibility mode. A list of the IEAIPS member that was used is in Appendix A, "IEAIPS Member used in the WLM Test Cases" on page 107. Also, the IEAICS and IEAOPT members can be found in Appendix B, "IEAICS Member used in the WLM Test Cases" on page 111 and Appendix C, "IEAOPT Member used in the WLM Test Cases" on page 119. Appendix D, "Service Definition — PERFPOL" on page 121 has a list of the service policies used in the test cases.

4.1 TSO Workload

The first test was done by the S/390 Division to investigate the ability of the WLM to manage a large TSO environment. The runs were made on an ES/9000 9021-982 (8-way) with 2048MB central storage and 2048MB expanded storage. The workload was the standard LSPR TSO workload with 3,750 logged-on users. See *Introduction to Large Processor Capacity and Performance Evaluation* for a description of the workload.

This test demonstrates the differences between running a large TSO system in goal mode and compatibility mode. In compatibility mode, the TSO workload was defined as having three periods. Each period is defined in the IPS. The importance of each period is not explicitly defined; for each period we specified the dispatching priority and the *domain* for the TSO users. The system dispatched the address spaces at the dispatching priority we specified, and managed the domains to the multiprogramming level within the constraints we specified.

In goal mode, we simply specified the relative importances of the three periods, and specified a response time objective.

4.1.1 TSO goals

The goals we set for the TSO users were based on the results observed running in compatibility mode. If you already have a formal service level agreement, you could use those values to set the goals. We recommend either an average response time or a response time with percentile goal for TSO users.

The following is a fragment of the RMF report taken in compatibility mode.

WORKLOAD ACTIVITY														PAGE	
MVS/ESA		SYSTEM ID UNKN		DATE 05/23/94		INTERVAL 30.00.132		MODE=COMPAT				1			
SP5.1.0		RPT VERSION 5.1.0		TIME 17.38.30		IPS = IEAIP5B									
OPT = IEAOPTGM		REPORT BY PERFORMANCE GROUP				SERVICE DEFINITION COEFFICIENTS				SU/SEC=2323.8					
ICS = IEAICS51		PERIOD				IOC = 5.0 CPU = 10.0 SRB = 10.0 MSO = 3.0000									
PGN	PGP	DMN	TIME SLICE GROUP	INTERVAL SERVICE	AVERAGE ABSORPTION, AVG TRX SERV RATE, TCB+SRB SECONDS, %		PAGE-IN RATES	STORAGE	TRANSACTIONS	AVG TRANS. TIME, STD. DEVIATION HHH.MM.SS.TTT					
SUBSYS = TSO		TRXCLASS =		ACCTINFO = NO											
USERID =		TRXNAME =		SRVCLASS =											
0020	1	020	**	IOC	9240K	ABSRPTN	7,801	SINGLE	7.51	AVERAGE	277.16	AVG	21.93	TRX	000.00.00.060
				CPU	103.6M	TRX SERV	7,336	BLOCK	3.58			MPL	20.63	SD	000.00.00.166
				MSO	169.0M	TCB	4,459.3	HSP	0.00	TOTAL	5,718.3	ENDED	379,200		
				SRB	7828K	SRB	336.8	HSP MISS	0.00	CENTRAL	4,896.0	END/SEC	210.65	QUE	000.00.00.000
				TOT	289.7M	TCB+SRB%	266.4	EXP SNGL	55.29	EXPAND	822.34	#SWAPS	378,148		
				PER SEC	160,951	EX VEL%	31.2	EXP BLK	0.02					TOT	000.00.00.060
0020	2	021	**	IOC	3308K	ABSRPTN	8,939	SINGLE	3.14	AVERAGE	319.27	AVG	6.64	TRX	000.00.00.431
				CPU	34893K	TRX SERV	8,939	BLOCK	0.67			MPL	6.64	SD	000.00.00.366
				MSO	67478K	TCB	1,501.5	HSP	0.00	TOTAL	2,121.8	ENDED	27,983		
				SRB	1271K	SRB	54.6	HSP MISS	0.00	CENTRAL	1,902.0	END/SEC	15.54	QUE	000.00.00.013
				TOT	106.9M	TCB+SRB%	86.4	EXP SNGL	5.97	EXPAND	219.77	#SWAPS	27,986		
				PER SEC	59,412	EX VEL%	30.9	EXP BLK	0.00					TOT	000.00.00.444
0020	3	022	**	IOC	20363K	ABSRPTN	5,553	SINGLE	0.51	AVERAGE	391.17	AVG	24.51	TRX	000.00.02.050
				CPU	65874K	TRX SERV	5,553	BLOCK	0.18			MPL	24.51	SD	000.00.02.326
				MSO	155.2M	TCB	2,834.7	HSP	0.00	TOTAL	9,590.8	ENDED	29,209		
				SRB	3652K	SRB	157.1	HSP MISS	0.00	CENTRAL	8,887.7	END/SEC	16.22	QUE	000.00.00.007
				TOT	245.1M	TCB+SRB%	166.1	EXP SNGL	1.16	EXPAND	703.03	#SWAPS	27,995		
				PER SEC	136,169	EX VEL%	13.8	EXP BLK	0.00					TOT	000.00.02.058
0020	ALL	ALL	ALL	IOC	32911K	ABSRPTN	6,883	SINGLE	3.64	AVERAGE	336.53	AVG	53.10	TRX	000.00.00.217
				CPU	204.4M	TRX SERV	6,714	BLOCK	1.59			MPL	51.79	SD	000.00.00.802
				MSO	391.8M	TCB	8,795.6	HSP	0.00	TOTAL	17,431	ENDED	436,392		
				SRB	12751K	SRB	548.7	HSP MISS	0.00	CENTRAL	15,686	END/SEC	242.42	QUE	000.00.00.001
				TOT	641.8M	TCB+SRB%	519.0	EXP SNGL	23.34	EXPAND	1,745.2	#SWAPS	434,129		
				PER SEC	356,533	EX VEL%	23.0	EXP BLK	0.01					TOT	000.00.00.219

Figure 38. TSO Workload in Compatibility Mode. Part of the RMF Workload Activity report is shown.

Figure 38 shows the workload activity report for the TSO users in periods 1 to 3. The total response time (TOT) was used as a guideline for the TSO response time goal. This is because in goal mode the average transaction time includes queuing time. In TSO the queuing time normally is zero except when logging on. The total response time for periods 1 to 3 is 0.060, 0.444, and 2.058 seconds, respectively.

The following is a fragment of the goal mode RMF report for comparison purposes.

WORKLOAD ACTIVITY										PAGE 20
MVS/ESA SP5.1.0	SYSPLEX PLEX1 RPT VERSION 5.1.0	DATE 06/12/1994 TIME 23.55.22	INTERVAL 30.00.188	MODE = GOAL						
POLICY ACTIVATION DATE/TIME 06/12/1994 23.29.29										
----- SERVICE CLASS(ES)										
REPORT BY: POLICY=PERFDEF3	WORKLOAD=TSO	SERVICE CLASS=TSO	RESOURCE GROUP=*NONE	TSO transactions						
-TRANSACTIONS--	TRANSACTION TIME	HHH.MM.SS.TTT	INTERVAL	SERVICE	--SERVICE RATES--	-PAGE-IN RATES--	----STORAGE----			
AVG 59.40	ACTUAL	000.00.00.245	IOC	33157K	ABSRPTN 6189	SINGLE 0.00	AVG 348.76			
MPL 57.37	QUEUED	000.00.00.001	CPU	206727K	TRX SERV 5978	BLOCK 0.00	TOTAL 20008.3			
ENDED 435090	EXECUTION	000.00.00.243	MSO	385316K	TCB 8895.7	HSP 0.00	CENTRAL 16907.6			
END/SEC 241.69	STANDARD DEVIATION	000.00.01.022	SRB	14005K	SRB 602.6	HSP MISS 0.00	EXPAND 3100.78			
#SWAPS 433100			TOT	639204K	TCB+SRB% 528	EXP SNGL 0.02				
EXECUTD 0			PER SEC	355075		EXP BLK 0.00				
PER IMPORTANCE	PERF	--TRANSACTIONS--	-----RESPONSE TIME-----			-EX VEL%-	-CPU--	-EXE--		
	INDX	-NUMBER-	-%	-----GOAL-----	---ACTUAL---	TOTAL	GOAL	ACT	USING%	DELAY%
1	HIGH	0.1	377122	87	00.00.00.400	AVG 00.00.00.059	27.1	0.1	0.2	
2	MEDIUM	0.4	28517	7	00.00.01.000	AVG 00.00.00.405	45.8	17.5	20.7	
3	LOW	0.5	29451	7	00.00.05.000	AVG 00.00.02.468	8.7	5.2	54.3	
TOTAL			435090	100						

Figure 39. TSO Workload in Goal Mode. Part of the RMF Workload Activity report is shown.

Table 7 summarizes the goals we set in our policy.

Workload Period	Type of Goal	Goal	Importance
TSO P.1	Avg. Resp.	0.100 sec	HIGH
TSO P.2	Avg. Resp.	1.000 sec	MEDIUM
TSO P.3	Avg. Resp.	3.000 sec	LOW

4.1.2 Results

Table 8 on page 84 summarizes the results of the two runs.

<i>Table 8. TSO Test Results</i>			
	Compatibility Mode	Goal Mode	Delta
CPU busy	91.01%	93.35%	+2.57%
Transaction Rate	242.42	241.69	
Tran rate at 100% (ITR)	266.37	258.91	-2.8%
P1 response (avg.)	0.060	0.059	
P1 response (S.D.)	0.166	0.137	
P1 transaction rate	210.65	209.49	
P2 response (avg.)	0.431	0.405	
P2 response (S.D.)	0.366	0.302	
P2 transaction rate	15.54	15.84	
P3 response (avg.)	2.050	2.468	
P3 response (S.D.)	2.326	3.115	
P3 transaction rate	16.22	16.36	
Total response (avg.)	0.217	0.245	
Total response (S.D.)	0.802	1.022	
Total transaction rate	242.42	241.69	-0.3%
UIC min	7	3	
UIC max	80	53	
UIC avg	32	18	
Migration age min	505	751	
Migration age max	552	882	
Migration age avg	529	839	
ES writes	2998.01	2690.31	
ES reads	2080.19	1932.62	
ES migration	662.16	250.31	
AUX swap rate	2.96	5.26	
AUX page in	205.05	118.16	
AUX page out	113.41	0.23	
TSO captured %	519%	528%	+1.7
TSO captured % per tran	2.14%	2.18%	+2.0

4.1.3 Observations

Comparing the two runs we can make a few observations.

The transaction rate was almost the same for the two cases; 242.42 transactions per second for compatibility mode and 241.69 for the goal mode run. The goal mode did take a little more CPU time. The utilization went from 91.01% in compatibility mode to 93.35% in goal mode. Calculating the ITR difference (CPU% per transaction) we see that the cost of goal mode is an increase in total processor busy per transaction is:

$$\frac{91.01}{242.42} \div \frac{93.35}{241.69} = .972$$

This is a 2.8% increase in processor time per transaction. However, the response for the first and second period TSO transactions is better than COMPAT mode.

The average response is slightly better, but what is more important is that the standard deviation of the response is better as well. This measures the variability of the response time. Taking the sum of the average and the standard deviation as an approximation of the 90th percentile response, there is a 13.3% improvement in the first period response time in goal mode and an 11.2% improvement in second period.

Comparing the other measured numbers is interesting. It is clear that goal mode uses more main storage. The *Unreferenced Interval Count* (UIC) is a measure of the age of the pages in central storage. The minimum, maximum and average values of UIC went from 7, 80 and 32 in compatibility mode to 3, 53 and 18 in goal mode, indicating that storage was under more stress in goal mode.

An interesting observation is the difference between the way the two systems handle management of the third period TSO. The compatibility mode system had only 2.96 swaps per second to auxiliary storage, while the goal mode system had 5.26. Again this could be attributed to an increased demand for storage in the goal mode system, but there is also a significant difference in the behaviour of the two systems. The compatibility mode system did 113.4 page outs per second to auxiliary storage, while the goal mode system did .23 page outs per second. It is clear that the goal mode system chooses a swap-out rather than a page-out to address storage shortages.

4.2 Another TSO Workload

This test was done by a group of Systems Engineers from Germany to measure MVS on various large processors. This test was run on a ES/9000 9021-982 (8-way) with 2048MB central storage and 2048MB expanded storage. The workload was an ISPF 4.1 program development workload with 1,360 logged on users. For this test TPNS ran on the same machine as the system under test. The scripts simulate TSO/ISPF 4.1 transactions and include editing and browsing of datasets and PDS members, copy activities and compilation and execution of assembler and cobol programs. CLISTS are invoked implicitly. A total of 900 different TSO transactions consisting of 16 cycles of LOGON, through different scenarios, and LOGOFF, were invoked through the entire simulation. We applied a fixed user think time of 10 seconds.

The following table, Table 9 on page 86 summarizes the goals we set for the workload in this measurement.

Table 9. Goal for Second TSO Measurement

Workload Period	Type of Goal	Goal	Importance
TSO P.1	95% Response	0.100 sec	HIGH
TSO P.2	95% Response	0.300 sec	MEDIUM
TSO P.3	90% Response	1.000 sec	MEDIUM
TPNS	Velocity	90	

4.2.1 Results

The following summarizes the results.

	Compat Mode	Goal Mode	Delta
CPU busy	89.44%	89.53%	+0.10%
Transaction Rate	194.63	194.56	
ITR	217.61	217.31	-.14%
TSO captured %	485.6%	489%	
TSO captured % per tran	2.49%	2.51%	
TPNS captured %	34.9%	35.9%	
P1 response (avg.)	0.012	0.023	
P1 response (S.D.)	0.011	0.034	
P1 transaction rate	99.86	119.67	
P2 response (avg.)	0.059	0.191	
P2 response (S.D.)	0.048	0.144	
P2 transaction rate	41.87	60.62	
P3 response (avg.)	0.333	0.643	
P3 response (S.D.)	0.274	0.515	
P3 transaction rate	52.89	14.34	
Total response (avg.)	0.110	0.117	
Total response (S.D.)	0.200	0.216	
Total transaction rate	194.63	194.56	

4.2.2 Observations

Comparing the two runs we can make a few observations.

Firstly, there is almost no cost for goal mode for this test. The reduction in ITR was only -.14%. We must take care comparing this run to TSO measurement discussed in 4.1, "TSO Workload" on page 81. The workload is different, using

entirely different scripts, and different policies. However, the hardware was the same, and both were heavy TSO environments.

One of the most obvious is the reduction in users. This experiment had only 1,360 users logged on, so there was no paging or swapping to DASD.

Secondly, the response is comparable between goal mode and compatibility mode. The average response degraded from .110 seconds to .117.

Note that the response time for each period is worse than compatibility mode, however, the transaction rate was much higher in first and second period. In compatibility mode, many of the transactions finished in third period.

4.3 Storage Stress Environment

This test was designed to examine WLM behaviour when storage is significantly over-committed.

This test is important because it demonstrates the system behaviour running an environment that would be very difficult, if not impossible, to run without the functions provided by the WLM running in goal mode.

A TSO workload consisting of 350 logged-on users was run on a 9021-942 with 512MB of central storage and 1024MB of expanded storage. After the TSO had run for a short while, four synthetic batch jobs were added to the system.

These batch jobs were set up to simulate large scientific compute-intensive jobs. Four jobs, requiring 1300MB, 600MB, 50MB and 50MB, were run. Each would have consumed 100% of the time available on a single CP, except for any delays caused by the paging resulting from the over-commitment of storage.

Table 10 summarizes the goals that we specified for the system. Note that the TSO workload was split into two service classes TSOODD and TSOEVEN based on the VTAM LU name. Both service classes had the same goals specified. This test did not distinguish between the different TSO communities, and so the results were combined in the following analysis.

The batch jobs were defined as discretionary, reflecting our objective of running the batch as well as possible without interfering with the TSO.

Workload Period	Type of Goal	Goal	Importance
TSOEVEN P.1	Avg. Resp.	0.300 sec	MEDIUM
TSOEVEN P.2	Avg. Resp.	1.500 sec	MEDIUM
TSOEVEN P.3	Avg. Resp.	5.000 sec	LOW
TSOODD P.1	Avg. Resp.	0.300 sec	MEDIUM
TSOODD P.2	Avg. Resp.	1.500 sec	MEDIUM
TSOODD P.3	Avg. Resp.	5.000 sec	LOW
BATCH	Discretionary	N/A	N/A

Figure 40 on page 88 shows the amount of service given to the various service classes. The X axis is the time of day, and the Y axis is the CPU service rate consumed by the various workloads. The two TSO service classes were added together to show the total CPU rate by each TSO period.

Note how relatively flat the service delivered to TSO remained throughout the measurement interval, including the time that the batch is running.

Note when the batch is released, the system delivers more and more CPU service to the batch, peaking at slightly more than 10,000 service units per second.

Around 1:25 AM the two small jobs ended, and the CPU service drops. At this time the remaining two batch jobs were left in the system competing for storage, causing significant paging. Around 1:34 AM, the 600MB job ended, leaving the 1300MB job as the only job in the discretionary service class.

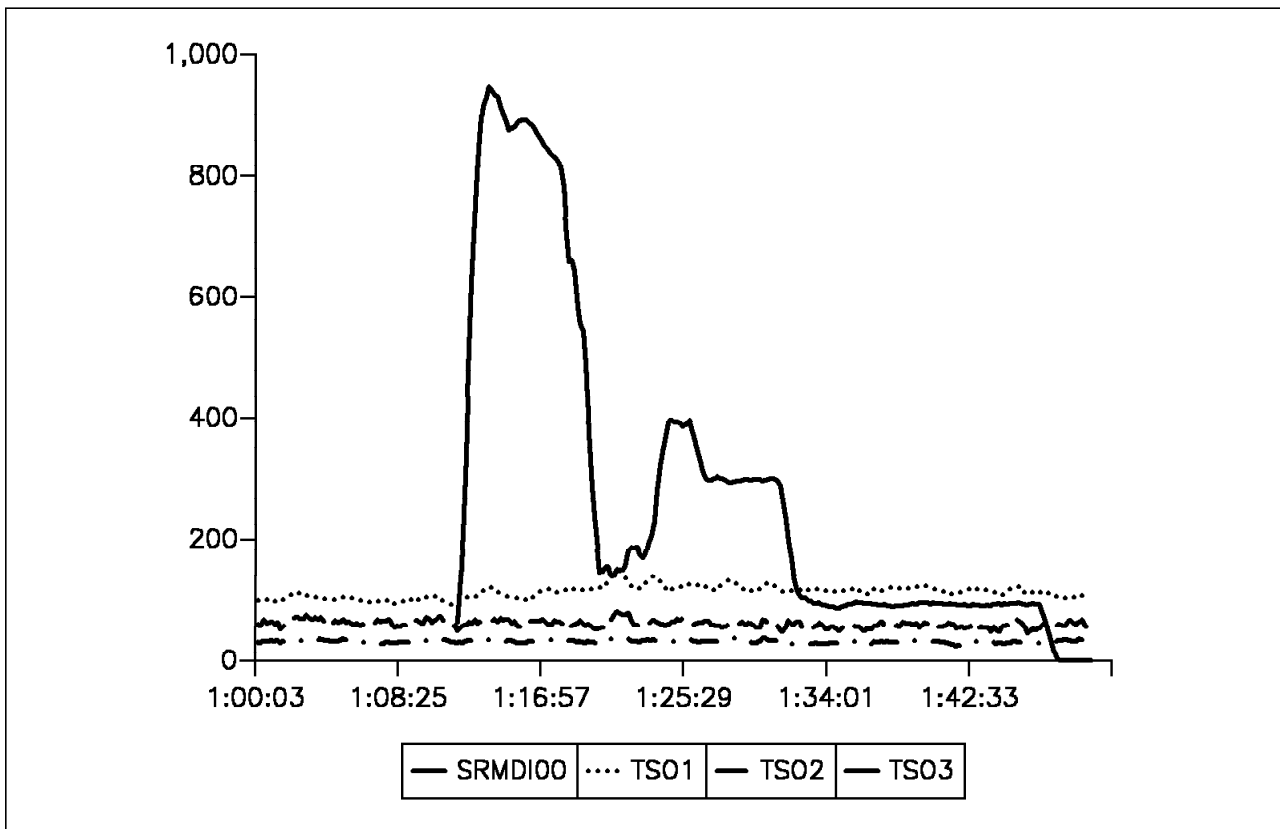


Figure 40. Batch Storage Stress. Consumed Service by Workload and Time.

While these “thrashers” were running, what happened to TSO? Nothing! Figure 41 on page 89 shows the performance index (PI) of the various service classes plotted against the time of day. Remember that the PI is the response time of the TSO users, normalized to the response time goal. Discretionary work has an artificially assigned PI of .81, indicating that it always is meeting its goal.

You can see on the graph where the discretionary work starts. Observe the behaviour of the TSO PI. Resources are stolen from TSO until the PI gets close to 1.0. If TSO is delayed for paging, WLM gives the TSO address spaces pages owned by the discretionary batch preventing the high paging stress from impacting the TSO response time.

Note that the PI for TSO exceeded 1.0 only at the start of the batch, and again briefly when the two small jobs ended. During job initiation and termination resources are consumed by the service classes SRMBEST and SRMGOOD, both of which are of higher importance than the TSO workload.

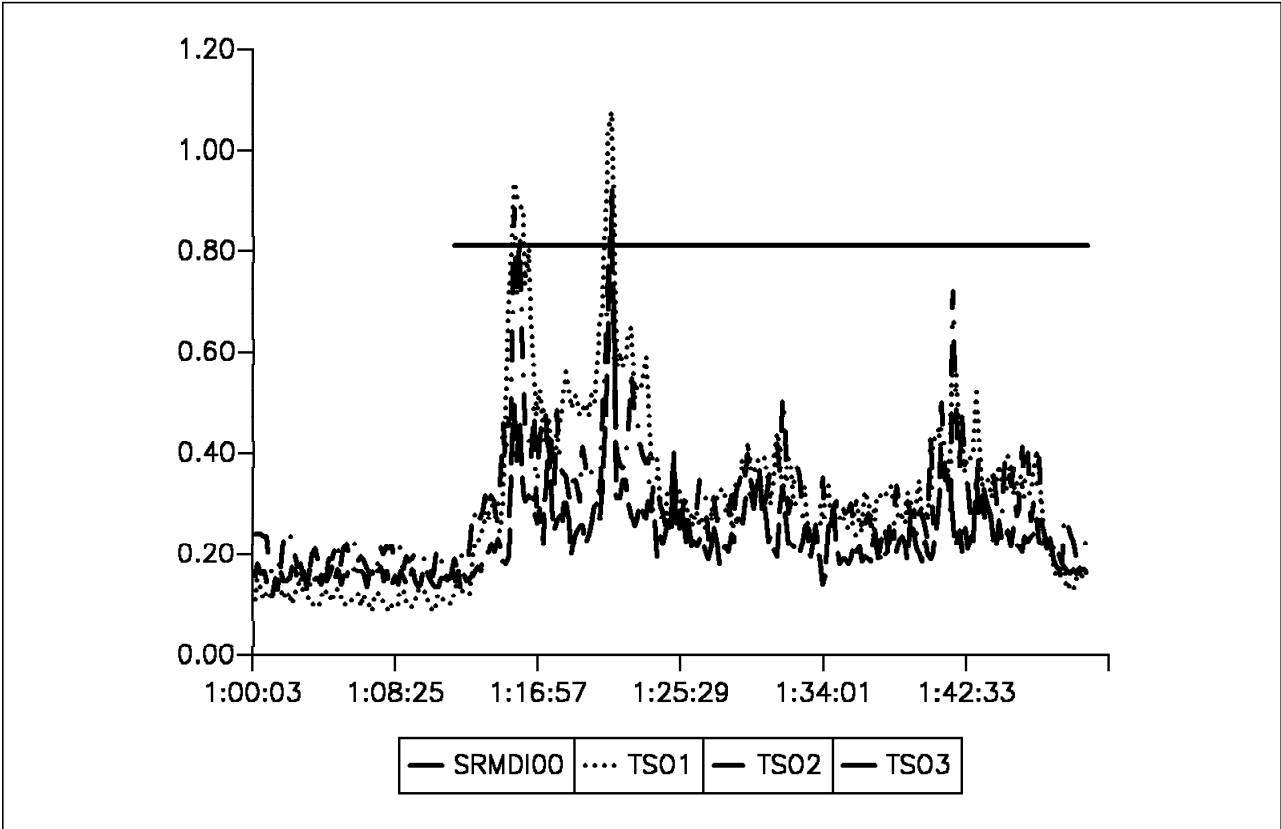


Figure 41. Batch Storage Stress. Performance Index over time.

This is an important test. Even though the discretionary batch was driving the system to very high paging rates, the TSO was not affected. The system had 19 local page data sets, each approximately 44% busy. The total page-in rate was 999.27 pages per second. See Figure 42 on page 90 for the RMF paging report.

PAGING ACTIVITY										PAGE 1	
MVS/ESA SP5.1.0	SYSTEM ID UNKN RPT VERSION 5.1.0				DATE 03/15/94 TIME 01.11.49	INTERVAL 30.00.356 CYCLE 1.000 SECONDS					
OPT = IEAOPT42											
CENTRAL STORAGE PAGING RATES - IN PAGES PER SECOND											
PAGE IN					PAGE OUT						
CATEGORY	SWAP	NON SWAP, BLOCK	NON SWAP, NON BLOCK	TOTAL RATE	% OF TOTL SUM	SWAP	NON SWAP	TOTAL RATE	% OF TOTL SUM		
PAGEABLE SYSTEM AREAS (NON VIO)											
LPA			0.01	0.01	0						
CSA			0.00	0.00	0	0.00	0.00	0.00	0		
SUM			0.01	0.01	0	0.00	0.00	0.00	0		
ADDRESS SPACES											
HIPERSPACE		0.00		0.00	0	0.00	0.00	0.00	0		
VIO		0.04		0.04	0	0.69	0.69	0.69	0		
NON VIO	105.60	862.18	31.43	999.21	100	39.61	131.50	171.11	100		
SUM	105.60	862.22	31.43	999.25	100	39.61	132.19	171.80	100		
TOTAL SYSTEM											
HIPERSPACE		0.00		0.00	0	0.00	0.00	0.00	0		
VIO		0.04		0.04	0	0.69	0.69	0.69	0		
NON VIO	105.60	862.18	31.44	999.22	100	39.61	131.50	171.11	100		
SUM	105.60	862.22	31.44	999.27	100	39.61	132.19	171.80	100		
PAGE MOVEMENT WITHIN CENTRAL STORAGE				49.64	PAGE MOVEMENT TIME %		0.0				
AVERAGE NUMBER OF PAGES PER BLOCK				86.6							
BLOCKS PER SECOND				9.96							
PAGE-IN EVENTS (PAGE FAULT RATE)				41.40							
PAGING ACTIVITY										PAGE 2	
MVS/ESA SP5.1.0	SYSTEM ID UNKN RPT VERSION 5.1.0				DATE 03/15/94 TIME 01.11.49	INTERVAL 30.00.356 CYCLE 1.000 SECONDS					
OPT = IEAOPT42											
EXPANDED STORAGE MOVEMENT RATES - IN PAGES PER SECOND											
ESF CONFIGURATION								HIGH UIC	MIGR AGE		
INSTALLED	ONLINE							MIN	0	88	
262144	262144							MAX	240	35287	
						AVG	55.7	4755.3			
		WRITTEN TO EXP STOR	READ FROM EXP STOR	MIGRATED FROM EXP STOR	FREED WITHOUT MIGRATION	*----- EXPANDED STORAGE FRAME COUNTS -----*					
						MIN	MAX	AVG			
TOTAL PAGES	RT %	6,564.44 100.0%	5,371.36 100.0%	986.31 100.0%	7.75	68,331	262,134	252,952			
HIPERSPACE PAGES	RT %	0.00 0.0%	0.00 0.0%	0.00 0.0%		0	0	0			
VIO PAGES	RT %	10.21 0.2%	1.98 0.0%	0.09 0.0%		0	126	62			

Figure 42. RMF Paging Activity Report for Storage Stress Environment

The paging statistics for the batch jobs are interesting. See Table 11 on page 91 for a summary.

Note that in this environment, the system depends almost exclusively on block paging rather than single demand paging. There were zero page-outs to auxiliary storage for all the jobs, and few page-ins. The 600MB job had much higher use of expanded storage than the 1300MB job; apparently it could fit. The 1300MB job had lower use of expanded storage and proportionally higher usage of the page data sets.

<i>Table 11. Paging Statistics for Batch Storage Stress Jobs</i>				
	T5000051	T5000052	T6000600	T1001300
TCB time	6:39.54	6:39.18	7:50.27	2:10.90
Page ins from Aux	73	55	49	174
Page outs to Aux	0	0	0	0
Page ins to ES	104,707	71,335	365,290	13,727
Page outs to ES	8,692	16,048	3,225	3,356
Block Pages in from Aux	2,910	2,340	284,749	1,555,064
Block pages out to Aux	3,495	1,694	91,080	137,634
Block pages in from ES	6,699	3,180	4,470,350	3,908
Block pages out to ES	117,861	70,665	5,176,255	1,674,029

4.4 Mixed Workload with CICS More Important

In this test case we modeled a heavy mixed online workload consisting of CICS transactions, IMS transactions, and TSO users running on a two-way 9021 at utilizations consistently higher than 99%. This scenario is very important, because it demonstrates something that is virtually impossible to do in compatibility mode.

Picture the prime shift of the compute centre, running three different sets of online users, at essentially 100% utilization. Now, someone shows up with a batch job stream that *has* to be run at the same time. How do you add a batch workload to a system running at 100% busy *without* disturbing the online response time?

In compatibility mode this is virtually impossible. Either the batch runs below the online, and is shut out from accessing their processor, or it runs with one of the online services, and takes more resource than it ought, and impacts the online response time. Given enough time, and some trial and error, a system programmer can design an IPS that uses *time slicing* that could probably solve this problem. However, it is a very difficult task.

With *goal mode* the service manager assigns goals, and an importance to each piece of work. In this example we have chosen to establish a policy that the CICS service is the most important to the data centre, that TSO and IMS are slightly less important, and that batch is of low importance.

Table 12 on page 92 summarizes the policy.

<i>Table 12. Goals for Mixed Workload with CICS More Important</i>			
Workload Period	Type of Goal	Goal	Importance
CICUSRTX	Avg. Resp.	0.090 sec	HIGH
CICSYSTX	Avg. Resp.	0.090 sec	HIGH
IMS	Avg. Resp.	10.000 sec	MEDIUM
IMS1	Avg. Resp.	0.180 sec	MEDIUM
TSO P.1	Avg. Resp.	0.100 sec	MEDIUM
TSO P.2	Avg. Resp.	1.000 sec	MEDIUM
TSO P.3	Avg. Resp.	3.000 sec	LOW
BATCHHI	Velocity	7%	LOWEST
BATCHLOW	Velocity	1%	LOWEST

Figure 43 on page 93 shows the “Consumed Service by Workload.” The Y axis shows the amount of service units consumed by each of the four different workloads running on the system. The X axis is time of day.

You can see that the three online workloads were all consuming a little more than 10,000 service units each, and that there is no batch until 7 or 8 minutes into the run.

Figure 44 on page 94 shows the performance indexes of the various workloads over the same time frame.

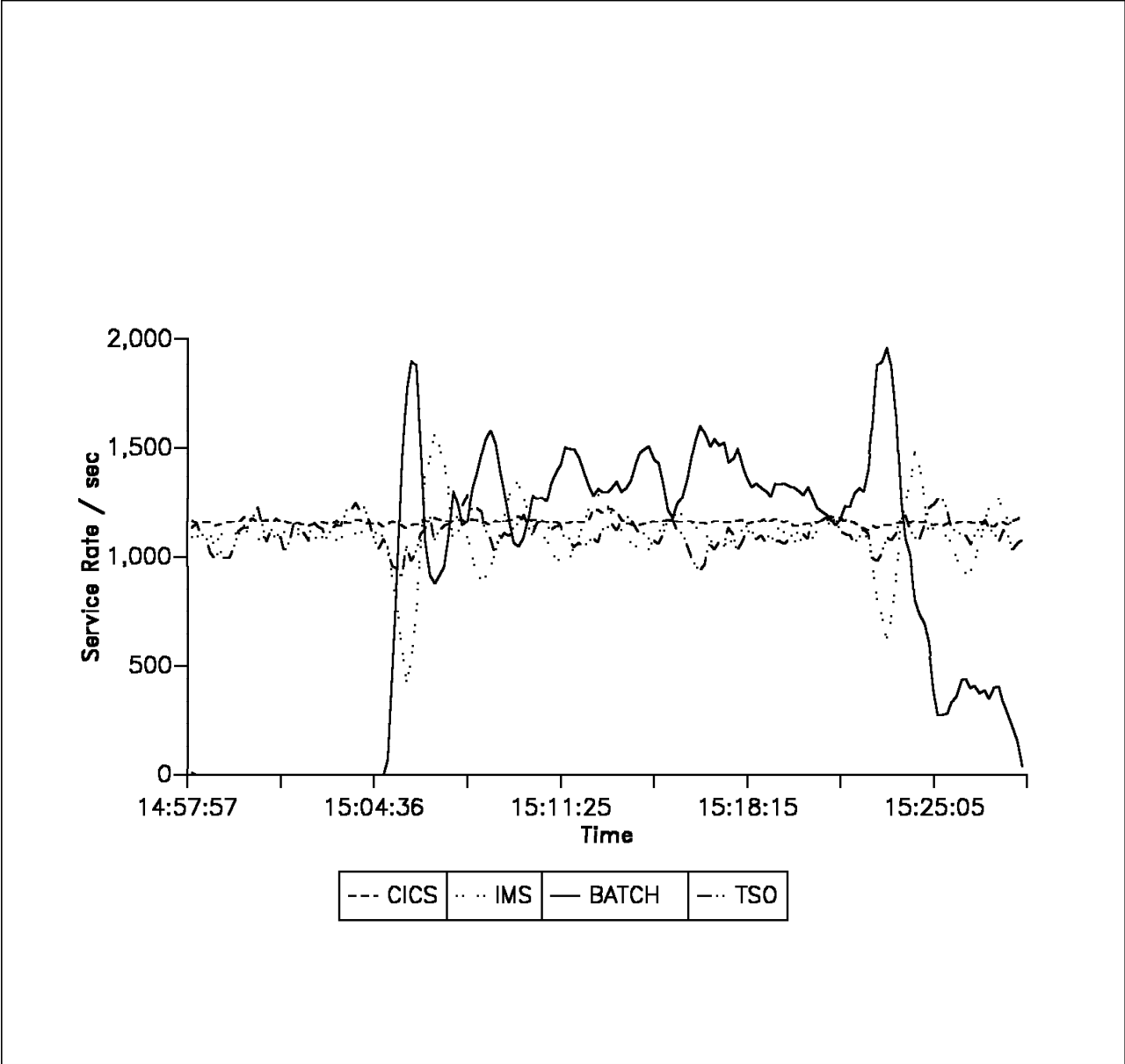


Figure 43. CICS More Important Test Case. Consumed service by workload and time.

Performance Index

CICS more important

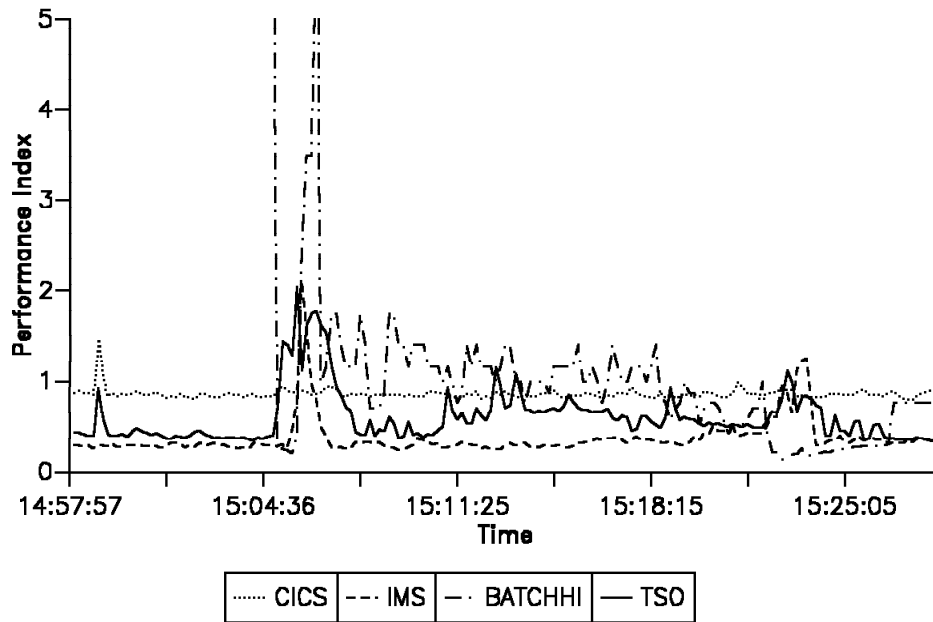


Figure 44. Performance Index for CICS Important Test Case

Now let's take a close look at the first few minutes in Figure 43 on page 93. Examine the amount of service given to each of the three online workloads. They are all very close together, indicating that this system is delivering the same amount of CPU service to each of three sets of online users. There is no batch at this time.

Now look at the same time frame on Figure 44. With the exception of a little blip at 15:00:00, the PIs are quite stable, and less than one. This means that the system is meeting the goals of all the online users. What happened at 15:00:00? RMF and SMF were synchronized to end their intervals on the hour. These started tasks were allowed by the classification rules to run in *Internal Service Class* "\$SRMG00D," which is more important than any of the user work. See 5.2.4, "Started Tasks and Other System Address Spaces" on page 101 for a discussion on the considerations for classifying started tasks. Remember that we were running at 100% in this test; a customer may or may not be able to detect this kind of a blip if running at lower utilizations.

Now, what happens to the system when the batch arrives? Notice the service units consumed rising quickly just after 15:04:36. So batch is getting CPU resources; who is the *donor*? We can tell from the same graph that the service delivered to IMS drops rapidly, concurrent with the rise in batch. Why did WLM choose IMS as the donor? Check the PI graph. IMS had the best PI (around .2) and was doing much better than its goal — it is the obvious candidate. Notice that TSO was a donor as well; its service dropped, but not as far as IMS. How long did we deliver service to the batch? Checking the PI graph, we favoured

batch until IMS and TSO had PIs of almost two. At that time WLM realizes that the online services are missing their goals, and takes resources from the batch.

The IMS PI drops very quickly back to almost its previous value, and TSO drops more slowly. Why more slowly? Taking a few resources from IMS caused a very large effect in the PI and TSO did not have such a great reaction. TSO's third period is also less important than IMS.

Notice that until the batch ended, the IMS PI stayed near its old value, and TSO oscillated just under one, but higher than its previous value. During this time the online services are all meeting their goals, and the batch is running.

This is an important function of WLM in goal mode. We can "squeeze" resource out of a 100% loaded system to run less important work and still meet the goals of the important workload.

What happens to CICS during all this? Nothing! CICS is the most important; the batch had virtually no effect on its PI.

As a last point, notice what happens when the batch ends. As the initiators went idle, they went into Internal Service Class "\$SRMG00D," and competed for resources with the online services.

Let us now have a closer look at the point when the batch work just came in and most of the performance indexes are high, at 15:06:28 (see Figure 44 on page 94).

```

94.167 15:06:28 PS1 -----
GROUP  CLASS  P ADDRSP  SPI  LPI  GSR  RID CODE ACTION
$SRMS01E 1 M1A2    85   85   201 7530 wsm_end_a2b_cnt
$SRMS01E 1 M1A2    86   86   204 6510 hsk_sl_dec_ici_tar
IMS      1          129 129   204 240 pa_grec_cand
$SRMS020 1          129 129   204 280 pa_rrec_cand
$SRMS01E 1          86   86   204 880 pa_pro_rdon_cand
$SRMS020 1          129 129   204 620 pa_pmuo_rec
$SRMS01E 1          86   86   204 940 pa_pro_unc_don
CICUSRTX 1          86   86   204 933 pa_pro_served_gdon
$SRMS01F 1          86   86   204 960 pa_pro_unc_sec_don
CICUSRTX 1          86   86   204 933 pa_pro_served_gdon
CICSYSTX 1          20   20   204 933 pa_pro_served_gdon
$SRMS020 1          129 129   204 750 pa_pro_incp_rec
IMS1     1          36   36   204 936 pa_pro_served_grec
IMS      1          129 129   204 936 pa_pro_served_grec

```

Figure 45. SMF 99 Trace for CICS Important Workload. This is at a time when the batch workload came in and the performance indexes are relatively high.

- Figure 45 shows that the workload manager ends the interval where pages that will not hurt are taken from the internal service class \$SRMS01E which represents CICS transactions (7530 *wsm_end_a2b_cnt*).
- Then for the CICS region M1A1 in the same service class, the protective central storage target is decremented (6510 *hsk_sl_dec_ici_tar*).
- Service class IMS is identified as a goal receiver candidate (240 *pa_grec_cand*). IMS currently has a performance index of 1.29. Note that

batch is not considered a receiver even when its performance index is worse, because it is not as important.

- IMS is an external service class for which the goals were defined but internally it is represented by service class \$SRMS020. The address spaces in this service class are the actual resource receiver candidates (280 *pa_rrec_cand*).
- The internal service class \$SRMS01E has the best performance index of 0.86 and is considered a resource donor candidate (880 *pa_pro_rdon_cand*).
- WLM assesses a move up in priority for \$SRMS020 (620 *pa_pmuo_rec*).
- This would not significantly change the donor \$SRMS01E (940 *pa_pro_unc_don*).
- The internal service class \$SRMS01E represents the external service class CICUSRTX (933 *pa_pro_served_gdon*).
- But the internal service class \$SRMS01F is also executing CICUSRTX transactions and is identified as a secondary donor (960 *pa_pro_unc_sec_don*).
- \$SRMS01F represents the service classes CICUSRTX and CICSYSTX which are the goal donors (933 *pa_pro_served_gdon*).
- WLM came to the conclusion that a rise in priority for the receiver \$SRMS020 is the best thing to do (750 *pa_pro_incp_rec*).
- The service classes that benefit from this are IMS and IMS1 (936 *pa_pro_served_grec*).

From this example one could see that to help an external service class the Workload Manager identifies the internal service classes and helps the address spaces that are within the service classes.

Results of the CICS More Important Test Case

The workload with the highest importance gets help first when the goal is not met.

4.5 Capping Test Cases

The capping test runs were done on an ES/9000 9021-921 (two-way) processor with 512MB central and 1024MB expanded storage. In all test cases the CPU is 99.2% busy.

4.5.1 Test Case with a Maximum

In this test case a pure TSO workload was used. The TSO workload was divided into two, based on the LUNAME. One half, service class TSOEVEN, was in no resource group and hence, was not capped. The other half of the TSO workload in service class TSOODD was capped and was only allowed to consume 1500 unweighted CPU SU per second.

This workload consumes about 2000 SU/sec if it were unconstrained. With a capping value of 1500 CPU SU/sec we are below what the TSOODD service class could absorb and WLM must take action to limit the address spaces consumption of resource.

TSO Service

TSO Odd capped at 1500 SU/sec

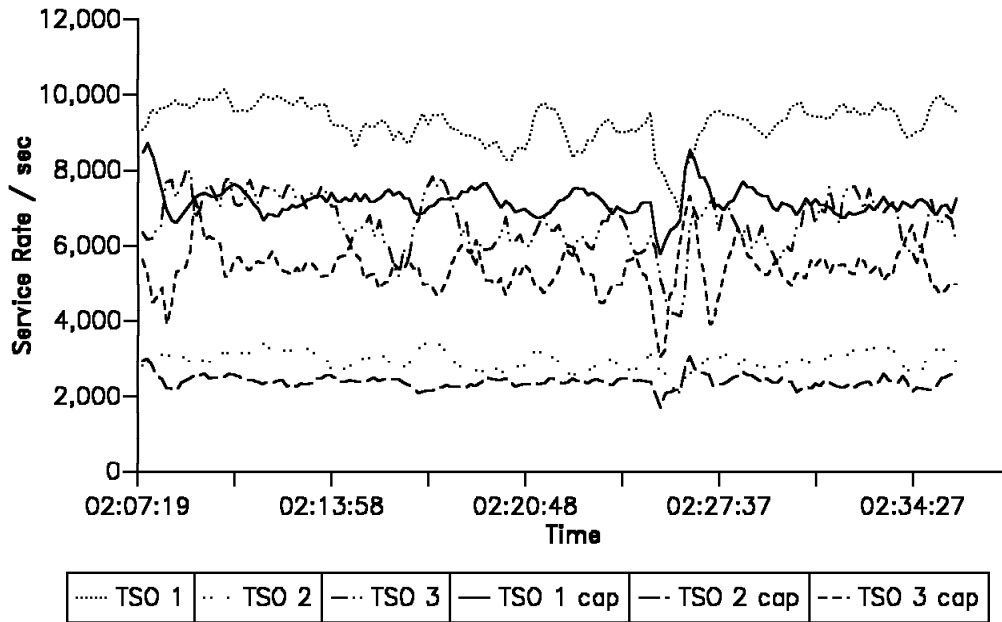


Figure 46. Consumed Service by Workload and Time. Test Case with TSO Odd capped by 1500 SU/sec.

As can be seen from Figure 46, the service in all TSO EVEN periods is higher than TSO Odd.

TSO Performance Index

TSO Odd capped at 1500 SU/sec

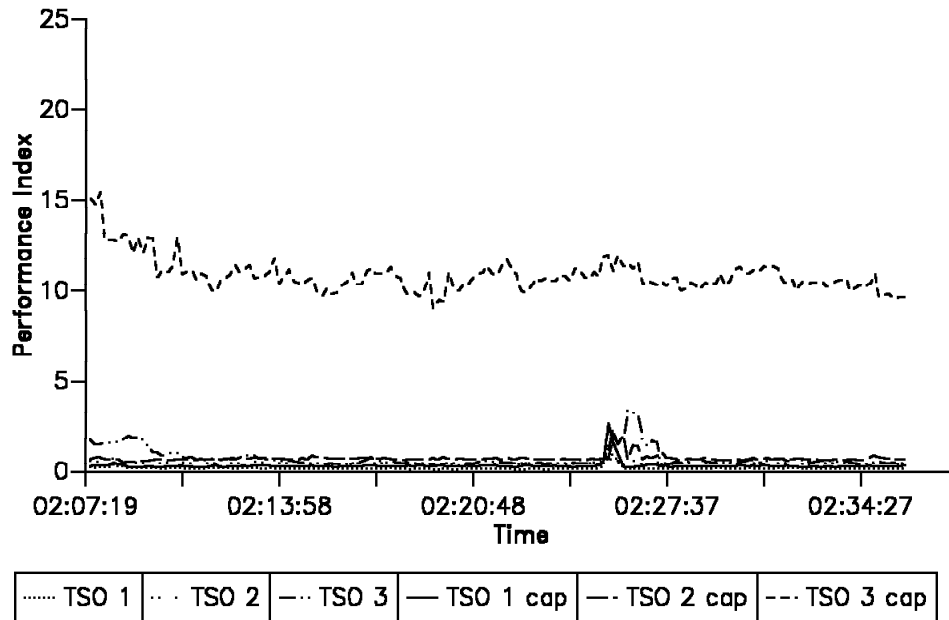


Figure 47. TSO Performance Index by Time. Test Case with TSO Odd capped by 1500 SU/sec.

It is not possible to reach the goals for all the TSO periods. TSO Period 3 in this test case is defined in the policy as the least important — These address spaces get hurt by the capping (see Figure 47). Its performance index is well above 1.0 and TSO period 3 users get longer response times. Remember that the PI is the response time normalized to the goal. The goal for third period was TSO was three seconds. A PI of 15 means that users in this period were seeing a response time of 60 seconds. This is probably not a reasonable TSO service, but remember that the least important service classes are delayed to reduce the consumption of a capped resource group. If there were batch service classes in the resource group, then the TSO service would have been unaffected, and the batch delayed.

Capping of a resource group may not apply to every customer environment, but it is a new function of WLM goal mode, and can be a useful management control over the workload.

Chapter 5. Migration Hints and Tips

In this chapter we will give some advice on how to migrate to goal mode and how to set up the goals. This is only a short summary. For more information about the implementation of workload management in goal mode refer to *MVS/ESA Planning: Workload Management*.

5.1 Software Positioning

- The first step is the migration to MVS/ESA Version 5. You do not have to change the IEAIPSxx, IEAICSxx, and IEAOPTxx members. The system will still run in compatibility mode as long as you specify IPS= in the IEASYSxx PARMLIB member at IPL.
- You should also upgrade your CICS and IMS subsystems to a level that supports workload manager services.
- The next step is to set up the system as a monoplex or sysplex if you do not already have a sysplex. This requires the formatting and definition of couple data sets. Refer to *MVS/ESA Setting up a Sysplex* for more information.
- Negotiate service level agreements with your users and use these as goals.
 - If your service level agreement contains end-to-end response time you have to convert it to system response time.
 - If you have some kind of throughput type goal you have to convert it into either a response time or a velocity goal.
- The next step is to define service classes. A service class is similar to a performance group and you could define a service class for each performance group. However, it is a good idea to have only a few service classes. You should group your workload into groups and assign them a service class. Use the ISPF dialog to specify the service definitions.
 - CICS and IMS long-running transactions may distort RMF data if they are reported in the same class as ordinary transactions. You might therefore want to define separate service classes for these transactions.

5.2 How to Obtain Goals

Many installations do not have service level agreements. Therefore the current response time or execution velocity can be used as a goal.

In the following we will describe how to get these numbers when running in compatibility mode. The basis for this are RMF reports which should be available for a reasonable time period, let's say one week.

5.2.1 TSO Workload

For TSO the total response time for each period as it is reported in the RMF workload activity report can be used as a guideline for a goal. To make migration easy, you should define the same number of periods in goal mode as in compatibility mode. You can define other goal types for TSO but an average response time goal will be the most appropriate if you are using existing RMF data to establish the goal, since RMF reports only average response times if the system is not running in goal mode.

5.2.2 Batch Workload

You will probably need more than one service class for your batch workload, maybe one for your important jobs and another one for the less important ones.

- For a job class which executes short running batch jobs, you can define an average response time goal.

The response time goal for batch jobs represents the total time. It is the sum of JES queue time and the execution time of the entire job. Note, that this is different to compatibility mode where each job step may be a separate transaction (there is one transaction for the first step of a job plus one additional transaction for each step which runs in a different performance group via PERFORM=). Therefore you should take the total elapsed time from job submission to job completion as it currently is in compatibility mode as a guideline for the goal in goal mode.

Do not use a response time goal for batch unless there are an average of at least ten job endings in a thirty minute period, to provide the system with some history.

- For long running jobs it is better to define an execution velocity. Since RMF Version 5 now reports the execution velocity in the Workload Activity Report, you can set up report performance groups for certain batch jobs to get the current execution velocity of these jobs.

5.2.3 CICS and IMS Workload

For CICS and IMS workloads you can define report performance groups for CICS or IMS transactions.

- RMF reports a response time for these transactions in the workload activity report. Note, however, that in compatibility mode there can be multiple transactions per CICS message. The time spent in the TOR is a separate transaction, as is the time in the AOR, and each invocation of an FOR.

Therefore, to get a guideline for a response time goal you can use the response time reported by the TOR as long as it is the only region type in the performance group.

- If IMS or CICS are already at a level that supports the workload manager services, you could define report performance groups in your IEAICSxx member that represent the service classes.

List the service classes under the subsystem (CICS or IMS) and associate them with a report performance group, for example:

```
SUBSYS=CICS
  SRVCLASS=CICSYSTX,RPGN=1000
  SRVCLASS=CICUSRTX,RPGN=1001
```

Define a service definition data set and activate a policy. This can be done with the command: VARY WLM,POLICY=polname.

Activating a policy does not mean your system is running in goal mode. It is still in compatibility mode but the workload manager is now aware of your service classes and goals.

- For CICS or IMS transactions you can only define one period.
- You can only define an average response time goal or a response time goal with percentile for a CICS or IMS transaction. CPSM however, does not support percentile goals.

Now, what goals should be defined for the CICS or IMS regions that execute the transactions?

- Actually you do not have to care much about the CICS, IMS, and other system address spaces. Goals for an IMS region are ignored as soon as these address spaces connect to the WLM and are recognized as servers.
- Since the goal is only used during the startup (or restart!) phase you should define a high velocity goal, for example a velocity of 80%, to ensure a fast initialization phase.

5.2.4 Started Tasks and Other System Address Spaces

Deciding how to handle started tasks is probably the most difficult part of preparing your WLM policy. There are several approaches, each with its own advantages and disadvantages.

1. Don't even bother classifying any started tasks

Suppose your WLM policy doesn't have any classification rules to assign service classes under the STC subsystem type. SRM will recognize certain system address spaces when they are created (like GRS, SMF, CATALOG, MASTER, RASP, XCFAS, SMXC, CONSOLE, IOSAS, WLM) and put them into the SYSTEM service class.

All other started tasks will be put into the SYSSTC service class, and kept at a very high dispatching priority. This is nice for JES, VTAM, and other system address spaces. If any given started task is recognized as serving online work with a goal (CICS or IMS transactions), then that address space would be managed as necessary based on the OLTP goal.

This approach has the advantage of simplicity. But not all started tasks are well-behaved. So having them all in SYSSTC could allow a CPU-intensive started task to use a large amount of processor cycles. If your processor is lightly loaded, or in a 6-way, 8-way, or 10-way MP, this might still be desirable because that single task won't affect the ability of the remaining processors to attack the important work that have goals.

Of course, you can still gather the traditional RMF data for any given started task by writing a classification rule that just assigns a report class, not a service class, to that started task.

2. Collect STC into a small number of similar groups

This approach continues the idea of simplicity. It's probably very easy to list really important started tasks, and also to list medium-importance started tasks.

Creating about five groups of started tasks might be a very legitimate approach. Probably you'd want the bottom group to have just a discretionary goal. Then you could assign increasing velocity goals to the other groups.

Your STC classification rules can simply refer to a named group of started tasks (transaction name group), or you can list tasks individually if you intend to use report classes to obtain extra RMF data for any individual started task.

The highest group probably contains such tasks as JES and VTAM. If you would prefer to let SRM handle this group of tasks "well," you should consider letting these run in the SYSSTC service class. Let's go through an example to do that. Let's say your transaction name group is called HI_STC,

and it contains VTAM, JES, LLA and RMF. In addition, suppose you've created a group called MED_STC as well, containing the group of less-favored started tasks.

The following example shows how the high started tasks are allowed to run in the SYSSTC service class:

```

Subsystem Type . . . . . : STC
Description . . . . . : All started tasks

-----Qualifier-----
Type      Name      Start
1 TNG      HI_STC   ___
1 TNG      MED_STC  ___
1 TN       *        ___

-----Class-----
Service   Report
DEFAULTS: _____
          _____
          VEL35  _____
          DISC  _____

```

The default service class is left blank. The first rule assigns no service class to transaction name group "HI_STC," so it inherits that blank default. This allows those started tasks in the group HI_STC to run in the SYSSTC service class.

The second rule assigns the tasks you identified as medium into a service class called VEL35. And the last rule says everything else receives a service class called DISC. SRM recognizes that this last rule shouldn't apply to the standard system address spaces (such as GRS, DUMPSRV, MASTER, WLM, etc.) and will continue to run them in the appropriate SYSTEM or SYSSTC service class. Or you could classify them yourself into a service class just like you would any other started task.

Remember, you can define a classification rule for any of the system started tasks mentioned above. But you are probably better off letting SRM control them by default in either the SYSTEM or SYSSTC service classes.

3. Classify most STC to individual service classes

That brings us to the last approach for handling started tasks. With this approach, each major started task is put into its own service class, similar to the way many accounts today have each STC in its own performance group. Thus you have service classes with names like VTAM, JES, LLA, VLF, etc.

One significant drawback to this approach is that each additional service class uses more storage to hold accumulated data for SRM, WLM, RMF, and SMF type 72 and type 99 records.

Another drawback is that SRM will be spending time trying to adjust these many service classes to achieve whatever velocity goals you create for them. As the discussion in 2.3, "Policy Adjustment" on page 34 states, SRM will help only one service class each time it assesses how well the system is addressing your policy. If you have a lot of service classes for individual started tasks, SRM could spend several intervals trying to handle an individual started task rather than addressing a problem facing your online or interactive work.

5.2.5 APPC

For APPC transactions the response time is reported in compatibility mode and can be used as a goal. But here again, the transaction time in compatibility mode is just the execution time.

To get a goal for the goal mode you should add the queue time of the initiator.

5.2.6 Capping

For some business reasons you might want to have a workload capped at some limited amount of CPU consumption. The workload should probably consist of different service classes, with different importances so that WLM can treat the different workloads appropriately. It is not normally recommended to cap an interactive workload but there may be good business reasons to do so.

- Define resource groups for the service classes you want to cap.
- Assign a capping value. Note that the capping value is in unweighted CPU and SRB service units per second and not the total amount of service. The capping value units are exactly the “SRM constant” units that have been reported by RMF on workload activity reports for years and are listed by processor model in the *MVS/ESA Initialization and Tuning Guide*.

5.3 Running in Goal Mode

With the system running in compatibility mode, with the policy active, we can set up report groups with the SRVCLASS parameter. This will allow us to get reports about how the response time or velocity would be for the service classes we have defined.

We can then adjust the goals or modify an importance of a workload before switching to goal mode.

The switch to goal mode can be done dynamically by the MODIFY WLM,MODE=GOAL command.

We can now monitor the system with RMF Monitor III and later on examine Monitor I postprocessor reports to see whether the goals were met or not.

You can switch to compatibility mode dynamically.

5.4 General Recommendations and Observations

Here is a list of recommendations and observations that can be helpful along your migration process. Some of them were already mentioned earlier in this book.

- If you have service class periods with goals and others just discretionary, it is likely that you will observe the ones with goals (even if not very important) will consistently exceed their goal (very small PI) with the discretionary not performing that well. This can happen because a discretionary service class period is never selected as a receiver. The one exception to this is when a discretionary service class is in a resource group which service rate is below the minimum.

- Do not define a velocity goal to a service class formed by CICS or IMS transactions. These transactions do not have an address space of their own and consequently their velocity cannot be measured.
- Do define a velocity goal to a service class formed by CICS or IMS regions. The goal for this service class will only be used until the address space is recognized as a server. Giving these regions a velocity goal ensure they get fast start-up.
- Do not define more than one period for a service class formed by CICS or IMS transactions. These transactions do not have an address space of their own and consequently the period migration cannot be detected.
- The compatibility mode external parameter DP (in IPS) should not be confused with the service definition *importance* keyword. They have distinct functions. For example, a high importance service class period does not always translate to a high dispatching priority. A service class period with a high importance goal which is easily achieved (for example TSO transactions with a response time goal of hours) may have a low dispatch priority relative to other work and still achieve its goal.
- The higher the CICS MAXTASK parameter (more transactions running in parallel), the more overhead in WLM due to PB sampling.
- Classify “system” work such as VTAM, JES to the SYSSTC service class. If system type of work is classified separately, SRM will try to manage it and it may take longer for other service classes to get resource adjustments. If you do decide to manage them separately, avoid the use of velocity above 80%.
- All the actions executed in the policy adjustment loop is logged every 10 seconds in an SMF 99 record. In unusual situations these records can help in tracing specific events. However, this recording uses lots of SMF DASD storage and it is not recommended that it be active in normal operation. You can use the T SMF=XX command to turn these records on and off. For general reporting and tuning information for a system in goal mode, you can use SMF 72 records (RMF workload management report).
- Do not be surprised if the dispatching priorities or storage isolation is not the same in goal mode as it was in compatibility mode. Workload Manager will manage the dispatching priorities and storage targets dynamically.

5.5 Summary

The intention of this book was to explain how the MVS/ESA Version 5 workload manager works and to show in some test cases that it really works that way.

The conversion to goal mode is quite easy and since you can always dynamically switch back to compatibility mode there is not much risk.

With goal mode it is much easier to set up a system in a way to get what you expected. There are no more strange parameters where you do not know what effect they have on the system behavior.

You can simply tell the Workload Manager what response time or velocity goal you expect from the system and the Workload Manager will do its best to reach the goal.

The Workload Manager is able to react much quicker to changes in the workload than a human being could do. No one changes an IEAIPSxx member every 10 seconds to change parameters to optimize the system.

The Workload Manager knows where the bottlenecks are and immediately takes actions to resolve a problem where a person would have to study reports for some time to find the problem.

So the Workload Manager could make even a well tuned system run more smoothly, providing more consistent response times according to the defined goals.

This should encourage you to migrate to goal mode and let the system do the tuning.

Appendix A. IEAIPS Member used in the WLM Test Cases

The following is the IPS member used in the WLM test cases.

```

/*****/
/*
/*          UPDATED IPS - BATCH, CICS, IMS, AND TSO          */
/*
/*
/*
/*****/
/*
/*          SYSTEM PERFORMANCE GROUP SPECIFICATIONS          */
/*
/* PERFORMANCE GROUP SPECIFICATIONS          RPGN SPECIFICATIONS */
/* -----          -----          */
/*
/* PGN=1, MISCELLANEOUS          101-130 CB80          */
/* PGN=2, DUMMY TSO PERF. GROUP          131-140 PCSV          */
/* PGN=10-19 BATCH          141-147 FPCX          */
/* PGN=11 HIGH PRIO & SORT          151-170 SCIE          */
/* PGN=20 TSO - PERFORMANCE          171-177 TXNCLASSES          */
/* PGN=30 IMS MCR          171 TXNCLASS A          */
/* PGN=31-34 IMS MPP' S - WFI (MPW)          172 TXNCLASS B          */
/* PGN=35-63 IMS MPP' S - OTHER          173 TXNCLASS G          */
/* PGN=64-69 CICS TXN REGIONS          174 TXNCLASS H          */
/* PGN=70 JES2 (RPGN=900)--          175 TXNCLASS J          */
/* PGN=71 JES3 (RPGN=901) |          176 TXNCLASS K          */
/* PGN=72 GTF (RPGN=902) |          177 TXNCLASS 4          */
/* PGN=73 SMS (RPGN=903) |          200-299 TSO          */
/* PGN=74 RMF & RMF811 (RPGN=904) |          300-399 IMS          */
/* PGN=75 COUNTS (RPGN=905) |          400-499 CICS REG 1          */
/* PGN=76 RSMON (RPGN=906) |          500-599 CICS REG 2          */
/* PGN=77 WLMON (RPGN=907) |          600-699 CICS REG 3          */
/* PGN=78 RTXA (RPGN=908) |          700-799 CICS DBDC          */
/* PGN=79 VTAM (NET) (RPGN=909) |          800-899 RESERVED          */
/* PGN=80 TCAS (RPGN=910) |          900-999 SYS & UTIL          */
/* PGN=81 SOF (RPGN=911) =====>          */
/* PGN=82 CHKLIST (RPGN=912) |          */
/* PGN=83 BR15 (RPGN=913) |          */
/* PGN=84 PAGING PGM (RPGN=914) |          */
/* PGN=85 JMON (RPGN=915) |          */
/* PGN=86 TCAM (RPGN=916) |          */
/*
/* NOTE:
/* ALLOCAS, CONSOLE, DUMPSRV, GRS, PCAUTH, TRACE, AND VFETCH
/* SHOULD NOT BE SPECIFIED SINCE THE MAJORITY OF THEIR SERVICE
/* UNITS ARE CHARGED TO THE CALLER. TO LIST THEM MAY PRESENT A
/* MISLEADING IMPRESSION OF THEIR ACTIVITY.
/*
/*****/
/*
/*          GENERAL SPECIFICATIONS          */
/*
/*****/
/*
CPU=10.0,IOC=5.0,MSO=3.0,SRB=10.0 /* DEFAULT COEFFICIENTS */
APGRNG=(0-15) /* FULL APG RANGE */
PVLDP=F82 /* DP FOR PERF. GROUP 0 */

```

```

TUNIT=1                               /* 1 SRM TUNIT PER SECOND */
/*
/*****/
/* EXCHANGE SWAP CONTROL IS SPECIFIED IN THE OPT W/ SWAPRSF PARM */
/* SHOULD BE SWAPRSF=10 (THE DEFAULT) */
/*****/
/*
/* DOMAINS DEFINITIONS: DOMAIN IMPORTANCE AND SERVICE SPECIFICATIONS*/
/*
/*****/
/*
DMN=1,CNSTR=(2,255)                    /* MISCELLANEOUS */
/*
DMN=10,CNSTR=(255,255),ASRV=(200,2500) /*BATCH, MPL = # INITS */
DMN=11,CNSTR=(255,255),DSRV=(1,99999999) /*SEP. DMN FOR HI PRI BTCH*/
/*
DMN=20,CNSTR=(160,255),ASRV=(3000,25000) /* TSO P1 W/ DUR=2K */
DMN=21,CNSTR=(160,255),ASRV=(650,3000) /* TSO P2 W/ DUR=2K */
DMN=22,CNSTR=(100,100)                 /* TSO P3 */
/*
DMN=30,CNSTR=(2,255)                   /* NON SWAP IMS MCR */
DMN=31,CNSTR=(2,255)                   /* IMS MPR' S - WFI */
DMN=35,CNSTR=(2,255)                   /* IMS MPR' S - OTHER */
/*
DMN=45,CNSTR=(2,255)                   /* DB2 */
DMN=46,CNSTR=(2,255)                   /* DB2 */
DMN=47,CNSTR=(2,255)                   /* DB2 */
/*
DMN=60,CNSTR=(1,1)                     /* IMS DBRC */
DMN=61,CNSTR=(1,1)                     /* IMS DLISAS */
/*
DMN=64,CNSTR=(3,255)                   /* CICS TXN REGIONS */
DMN=65,CNSTR=(3,255)                   /* CICS TXN REGIONS */
DMN=66,CNSTR=(3,255)                   /* CICS TXN REGIONS */
DMN=67,CNSTR=(3,255)                   /* CICS TXN REGIONS */
DMN=68,CNSTR=(3,255)                   /* CICS TXN REGIONS */
DMN=69,CNSTR=(3,255)                   /* CICS TXN REGIONS */
/*
DMN=70,CNSTR=(1,1)                     /* JES2 */
DMN=71,CNSTR=(1,1)                     /* JES3 */
DMN=72,CNSTR=(1,1)                     /* GTF */
DMN=73,CNSTR=(1,1)                     /* SMS */
DMN=74,CNSTR=(1,1)                     /* RMF & RMF811 */
DMN=75,CNSTR=(1,1)                     /* COUNTS */
DMN=76,CNSTR=(1,1)                     /* RSMON */
DMN=77,CNSTR=(1,1)                     /* WLMON */
DMN=78,CNSTR=(1,1)                     /* RTXA */
DMN=79,CNSTR=(1,1)                     /* VTAM (ACF/VTAM) */
DMN=80,CNSTR=(1,1)                     /* TCAS */
DMN=81,CNSTR=(4,4)                     /* SOF */
DMN=82,CNSTR=(4,4)                     /* CHKLIST */
DMN=83,CNSTR=(4,4)                     /* BR15 PROCEDURES */
DMN=84,CNSTR=(4,4)                     /* PAGING PROGRAM */
DMN=85,CNSTR=(1,1)                     /* JMON */
DMN=86,CNSTR=(4,4)                     /* TCAM */
/*****/
/* PERFORMANCE GROUPS DEFINITIONS */
/*****/
/*

```



```

PGN=1, (DMN=1,DP=M1)                /* MISCELLANEOUS                */
/*                                                                            */
PGN=2, (DMN=20,DP=F70,DUR=1K)        /* DUMMY PERF. GROUP            */
      (DMN=20,DP=F50,DUR=2K)        /* TO KEEP SYSTEM HAPPY        */
      (DMN=21,DP=M3)                /*                               */
/*                                                                            */
PGN=10, (DMN=10,DP=M2)               /* BATCH                        */
PGN=11, (DMN=11,DP=M5)               /* HIGH PRIO & SORT            */
/*=====*/
/*      TSO OBJECTIVE:                */
/*      90% (+/- 2) OF THE TXN'S FINISHING IN PERIOD 1                */
/*=====*/
/*                                                                            */
PGN=20, (DMN=20,DP=F70,DUR=2K)        /* TSO FIRST PERIOD            */
      (DMN=21,DP=F50,DUR=2K)        /* TSO SECOND PERIOD           */
      (DMN=22,DP=M3)                /* TSO THIRD PERIOD            */
PGN=30, (DMN=30,DP=F90)               /* IMS MCR                      */
PGN=31, (DMN=31,DP=F6)                /* IMS MPP' S - WFI            */
PGN=35, (DMN=35,DP=F6)                /* IMS MPP' S - OTHER          */
PGN=45, (DMN=45,DP=F74)               /* DB2                          */
PGN=46, (DMN=46,DP=F74)               /* DB2                          */
PGN=47, (DMN=47,DP=F92)               /* DB2                          */
PGN=60, (DMN=60,DP=F92)               /* IMS DBRC                     */
PGN=61, (DMN=61,DP=F90)               /* IMS DLISAS                   */
/*                                                                            */
PGN=64, (DMN=64,DP=F7)                /* CICS - 1ST REG              */
PGN=65, (DMN=65,DP=F7)                /* BCIC - 2ND REG              */
PGN=66, (DMN=66,DP=F7)                /* ACIC - 3RD REG              */
PGN=67, (DMN=67,DP=F7)                /* DBDC - STANDARD CICS        */
PGN=68, (DMN=68,DP=F7)                /* CICS TXN REGIONS            */
PGN=69, (DMN=69,DP=F7)                /* CICS TXN REGIONS            */
/*                                                                            */
PGN=70, (DMN=70,DP=F94)               /* JES2                         */
PGN=71, (DMN=71,DP=F94)               /* JES3                         */
PGN=72, (DMN=72,DP=F94)               /* GTF                          */
PGN=73, (DMN=73,DP=F81)               /* SMS                          */
PGN=74, (DMN=74,DP=F92)               /* RMF & RMF811                */
PGN=75, (DMN=75,DP=F91)               /* COUNTS                       */
PGN=76, (DMN=76,DP=F91)               /* RSMON                        */
PGN=77, (DMN=77,DP=F91)               /* WLMON                        */
PGN=78, (DMN=78,DP=F91)               /* RTXA                         */
PGN=79, (DMN=79,DP=F84)               /* VTAM (ACF/VTAM)            */
PGN=80, (DMN=80,DP=F83)               /* TCAS                         */
PGN=81, (DMN=81,DP=F80)               /* SOF                          */
PGN=82, (DMN=82,DP=F80)               /* CHKLIST                      */
PGN=83, (DMN=83,DP=F00)               /* BR15                         */
PGN=84, (DMN=84,DP=F93)               /* PAGING PROGRAM              */
PGN=85, (DMN=85,DP=F93)               /* JMON                         */
PGN=86, (DMN=86,DP=F84)               /* TCAM                         */

```


Appendix B. IEAICS Member used in the WLM Test Cases

The following is the ICS member used in the WLM test cases.

```

/*****/
/*                                          */
/*          SYSTEM PERFORMANCE GROUP SPECIFICATIONS          */
/*                                          */
/* PERFORMANCE GROUP SPECIFICATIONS          RPGN SPECIFICATIONS */
/* -----                                ----- */
/* PGN=1, MISCELLANEOUS                      101-130 CB84      */
/* PGN=2, DUMMY TSO PERF. GROUP             131-140 PCSV        */
/* PGN=10-19 BATCH                          141-147 FPCX        */
/* PGN=11 HIGH PRIO & SORT                  151-170 SCIE        */
/* PGN=20 TSO - PERFORMANCE                 171-177 TXNCLASSES */
/* PGN=30 IMS MCR                          171 TXNCLASS A     */
/* PGN=31-34 IMS MPP'S - WFI (MPW)         172 TXNCLASS B     */
/* PGN=35-63 IMS MPP'S - OTHER             173 TXNCLASS G     */
/* PGN=64-69 CICS TXN REGIONS              174 TXNCLASS H     */
/* PGN=70 JES2 (RPGN=900) --              175 TXNCLASS J     */
/* PGN=71 JES3 (RPGN=901) |              176 TXNCLASS K     */
/* PGN=72 GTF (RPGN=902) |              177 TXNCLASS 4     */
/* PGN=73 SMS (RPGN=903) |              200-299 TSO        */
/* PGN=74 RMF & RMF811 (RPGN=904) |      300-399 IMS        */
/* PGN=75 COUNTS (RPGN=905) |           400-499 CICS REG 1 */
/* PGN=76 RSMON (RPGN=906) |           500-599 CICS REG 2 */
/* PGN=77 WLMON (RPGN=907) |           600-699 CICS REG 3 */
/* PGN=78 RTXA (RPGN=908) |            700-799 CICS DBDC  */
/* PGN=79 VTAM (NET) (RPGN=909) |       800-899 RESERVED  */
/* PGN=80 TCAS (RPGN=910) |            900-999 SYS & UTIL */
/* PGN=81 SOF (RPGN=911) =====>      */
/* PGN=82 CHKLIST (RPGN=912) |          */
/* PGN=83 BR15 (RPGN=913) |            */
/* PGN=84 PAGING PGM (RPGN=914) |       */
/* PGN=85 JMON (RPGN=915) |            */
/* PGN=86 TCAM (RPGN=916) |            */
/* PGN=87 FSS SCAFFOLD (RPGN=917) |     */
/*                                          */
/* NOTE:                                          */
/* ALLOCAS, CONSOLE, DUMPSRV, GRS, PCAUTH, TRACE, AND VFETCH */
/* SHOULD NOT BE SPECIFIED SINCE THE MAJORITY OF THEIR SERVICE */
/* UNITS ARE CHARGED TO THE CALLER. TO LIST THEM MAY PRESENT A */
/* MISLEADING IMPRESSION OF THEIR ACTIVITY.     */
/*****/
SUBSYS=STC,PGN=1                               /* STARTED TASKS */
/*
TRXNAME=IMS(1),PGN=30                          /* IMS MCR          */
TRXNAME=MPW(1),PGN=31                          /* IMS MPP'S - WFI */
TRXNAME=MPP(1),PGN=35                          /* IMS MPP'S - OTHER */
TRXNAME=DBRC(1),PGN=60                         /* IMS DBRC         */
TRXNAME=DLISAS(1),PGN=61                      /* IMS DBRC         */
TRXNAME=DLIH(1),PGN=61                        /* IMS DBRC         */
TRXNAME=IRLM(1),PGN=62                       /* IMS DBRC         */
TRXNAME=CICS,PGN=64                          /* CICS 1ST REG     */
TRXNAME=BCIC,PGN=65                          /* CICS 2ND REG     */
TRXNAME=ACIC,PGN=66                          /* CICS 3RD REG     */
TRXNAME=DBDC,PGN=67                          /* STANDARD CICS    */
TRXNAME=JES2,PGN=70,RPGN=900                 /* JES2             */

```

```

TRXNAME=JES3,PGN=71,RPGN=901          /* JES3          */
TRXNAME=GTF,PGN=72,RPGN=902          /* GTF           */
TRXNAME=SMS(1),PGN=73,RPGN=903       /* SMS           */
TRXNAME=RMF(1),PGN=74,RPGN=904       /* ANY RMF       */
TRXNAME=COU(1),PGN=75,RPGN=905       /* COUNTS        */
TRXNAME=RSMON(1),PGN=76,RPGN=906     /* RSMON         */
TRXNAME=WLMON(1),PGN=77,RPGN=907     /* WLMON         */
TRXNAME=RTXA(1),PGN=78,RPGN=908     /* RTXA          */
TRXNAME=VTAM(1),PGN=79,RPGN=909     /* VTAM          */
TRXNAME=NET(1),PGN=79,RPGN=909     /* ACF/VTAM     */
TRXNAME=TCAS,RPGN=910                /* TCAS          */
TRXNAME=SOF(1),PGN=81,RPGN=911       /* SOF           */
TRXNAME=CHKLIST,PGN=82,RPGN=912      /* CHKLIST       */
TRXNAME=BR15(1),PGN=83,RPGN=913      /* BR15          */
TRXNAME=PAGING,PGN=84,RPGN=914       /* PAGING PROGRAM */
TRXNAME=JMON(1),PGN=85,RPGN=915      /* JOB MONITORING */
TRXNAME=TCAM(1),PGN=86,RPGN=916     /* TCAM          */
TRXNAME=PWTR(1),PGN=87,RPGN=917     /* FSS SCAFFOLD  */
TRXNAME=XA5MSTR,PGN=45                /* DB2           */
TRXNAME=XA5DBM1,PGN=46               /* DB2           */
TRXNAME=IRLMWFI,PGN=47               /* DB2           */
/*-----*/
/* SYSTEM ADDRESS SPACES REPORT GROUPS */
/*-----*/
TRXNAME=*MASTER*,RPGN=950
TRXNAME=ALLOCAS,RPGN=951
TRXNAME=CATALOG,RPGN=952
TRXNAME=CONSOLE,RPGN=953
TRXNAME=DUMPSRV,RPGN=954
TRXNAME=GRS(1),RPGN=955
TRXNAME=LLA(1),RPGN=956
TRXNAME=PCAUTH,RPGN=957
TRXNAME=SMF(1),RPGN=958
TRXNAME=TRACE,RPGN=959
TRXNAME=RASP,RPGN=960
TRXNAME=VLF(1),RPGN=961
TRXNAME=XCFAS,RPGN=962
TRXNAME=IOSAS,RPGN=963
TRXNAME=ANTMAIN,RPGN=965
TRXNAME=SMXC,RPGN=966
TRXNAME=SYSBMAS,RPGN=967
TRXNAME=WLM,RPGN=968
TRXNAME=INIT,RPGN=969
TRXNAME=DLF(1),RPGN=970              /* HIPERBATCH   */
TRXNAME=VFETCH,RPGN=971
TRXNAME=RACF(1),RPGN=973
TRXNAME=JESXCF(1),RPGN=974
/*-----*/
SUBSYS=TSO,PGN=20                    /* TSO           */
/*-----*/
TRXNAME=SPF(1),RPGN=200              /*              */
TRXNAME=ISPF(1),RPGN=270            /*              */
TRXNAME=PDF(1),RPGN=271             /*              */
TRXNAME=ASM,RPGN=201                /*              */
TRXNAME=COBOL,RPGN=202              /*              */
TRXNAME=PLIC,RPGN=203               /*              */
TRXNAME=EX(1),RPGN=204              /*              */
TRXNAME=ALLOC(1),RPGN=205           /*              */
TRXNAME=E(1),RPGN=206               /*              */

```

```

TRXNAME=DEL(1),RPGN=207                /* */
TRXNAME=SCRIPT,RPGN=208                /* */
/*-----*/
SUBSYS=JES2,PGN=10,OPGN=11            /* JES2 */
/*-----*/
/*          TRANSACTION NAMES FOR CBW2 JOBSTREAM          */
/*-----*/
TRXNAME=CARP(1),RPGN=210                /* CBW2 */
TRXNAME=COBB(1),RPGN=211                /* CBW2 */
TRXNAME=COB2(1),RPGN=212                /* CBW2 */
TRXNAME=CPGM(1),RPGN=213                /* CBW2 */
TRXNAME=DB2J(1),RPGN=214                /* CBW2 */
TRXNAME=DPST(1),RPGN=215                /* CBW2 */
TRXNAME=INUP(1),RPGN=216                /* CBW2 */
TRXNAME=PAYR(1),RPGN=217                /* CBW2 */
TRXNAME=SALE(1),RPGN=218                /* CBW2 */
TRXNAME=SLRP(1),RPGN=219                /* CBW2 */
TRXNAME=SORT(1),RPGN=220                /* CBW2 */
TRXNAME=TIMS(1),RPGN=221                /* CBW2 */
TRXNAME=WHSE(1),RPGN=222                /* CBW2 */
TRXNAME=NPGM(1),RPGN=223                /* CBW2 */
/* */
/*-----*/
/*- CICS TRANSACTIONS/REGIONS          -*/
/*-----*/
TRXNAME=M1T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M1A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M1A2,PGN=65                    /* CICS REG 2 */
TRXNAME=M1F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M2T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M2A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M2F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M3T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M3A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M3F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M4T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M4A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M4F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M5T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M5A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M5F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M6T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M6A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M6F1,PGN=66                    /* CICS REG 3 */
TRXNAME=M7T1,PGN=64                    /* CICS REG 1 */
TRXNAME=M7A1,PGN=65                    /* CICS REG 2 */
TRXNAME=M7F1,PGN=66                    /* CICS REG 3 */
TRXNAME=ECIC,PGN=64                    /* CICS REG 5 */
TRXNAME=FCIC,PGN=64                    /* CICS REG 6 */
TRXNAME=GCIC,PGN=64                    /* CICS REG 7 */
TRXNAME=HCIC,PGN=64                    /* CICS REG 7 */
TRXNAME=TCIC,PGN=64                    /* CICS TOR 1 */
TRXNAME=MCIC,PGN=65                    /* CICS AOR 1 */
TRXNAME=SCIC,PGN=66                    /* CICS FOR 1 */
/*-----*/
SUBSYS=M1T1,RPGN=101                    /* MRO PLEX #1 TOR #1 */
/*-----*/
TRXNAME=/FOR,RPGN=102
TRXNAME=XFOR,RPGN=103

```

TRXNAME=UFOR,RPGN=104
TRXNAME=WFOR,RPGN=105
TRXNAME=YFOR,RPGN=106
TRXNAME>ZFOR,RPGN=107
TRXNAME=HR1,RPGN=108
TRXNAME=HX1,RPGN=109
TRXNAME=HR2,RPGN=110
TRXNAME=HX2,RPGN=111
TRXNAME=DE1,RPGN=112
TRXNAME=DX1,RPGN=113
TRXNAME=SC2,RPGN=114
TRXNAME=SX2,RPGN=115
TRXNAME=SC4,RPGN=116
TRXNAME=SX4,RPGN=117
TRXNAME=SC6,RPGN=118
TRXNAME=SX6,RPGN=119
TRXNAME=TS1,RPGN=120
TRXNAME=TX1,RPGN=121
TRXNAME=OE1,RPGN=122
TRXNAME=OX1,RPGN=123
TRXNAME=OE2,RPGN=124
TRXNAME=OX2,RPGN=125
TRXNAME=OE4,RPGN=126
TRXNAME=OX4,RPGN=127
TRXNAME=OE5,RPGN=128
TRXNAME=OX5,RPGN=129
TRXNAME=IT1,RPGN=130
TRXNAME=IX1,RPGN=131
TRXNAME=IT2,RPGN=132
TRXNAME=IX2,RPGN=133
TRXNAME=IT8,RPGN=134
TRXNAME=IX8,RPGN=135
TRXNAME=PS2,RPGN=136
TRXNAME=PX2,RPGN=137
TRXNAME=PS3,RPGN=138
TRXNAME=PX3,RPGN=139
TRXNAME=DU1,RPGN=140
TRXNAME=DW1,RPGN=141
TRXNAME=DY1,RPGN=142
TRXNAME=DZ1,RPGN=143
TRXNAME=HR1A,RPGN=108
TRXNAME=HX1A,RPGN=109
TRXNAME=HR2A,RPGN=110
TRXNAME=HX2A,RPGN=111
TRXNAME=DE1A,RPGN=112
TRXNAME=DX1A,RPGN=113
TRXNAME=SC2A,RPGN=114
TRXNAME=SX2A,RPGN=115
TRXNAME=SC4A,RPGN=116
TRXNAME=SX4A,RPGN=117
TRXNAME=SC6A,RPGN=118
TRXNAME=SX6A,RPGN=119
TRXNAME=TS1A,RPGN=120
TRXNAME=TX1A,RPGN=121
TRXNAME=OE1A,RPGN=122
TRXNAME=OX1A,RPGN=123
TRXNAME=OE2A,RPGN=124
TRXNAME=OX2A,RPGN=125
TRXNAME=OE4A,RPGN=126

```

TRXNAME=OX4A,RPGN=127
TRXNAME=OE5A,RPGN=128
TRXNAME=OX5A,RPGN=129
TRXNAME=IT1A,RPGN=130
TRXNAME=IX1A,RPGN=131
TRXNAME=IT2A,RPGN=132
TRXNAME=IX2A,RPGN=133
TRXNAME=IT8A,RPGN=134
TRXNAME=IX8A,RPGN=135
TRXNAME=PS2A,RPGN=136
TRXNAME=PX2A,RPGN=137
TRXNAME=PS3A,RPGN=138
TRXNAME=PX3A,RPGN=139
TRXNAME=DU1A,RPGN=140
TRXNAME=DW1A,RPGN=141
TRXNAME=DY1A,RPGN=142
TRXNAME=DZ1A,RPGN=143
TRXNAME=H(1),RPGN=224
TRXNAME=D(1),RPGN=225
TRXNAME=S(1),RPGN=226
TRXNAME=T(1),RPGN=226
TRXNAME=O(1),RPGN=227
TRXNAME=I(1),RPGN=228
TRXNAME=P(1),RPGN=229
TRXNAME=CSPG,RPGN=144
TRXNAME=CSPQ,RPGN=145
TRXNAME=CSPS,RPGN=146
TRXNAME=C(1),RPGN=230
/*-----*/
SUBSYS=M1A1,RPGN=155          /* MRO PLEX #1 AOR #1          */
/*-----*/
TRXNAME=/FOR,RPGN=156
TRXNAME=UFOR,RPGN=157
TRXNAME=WFOR,RPGN=158
TRXNAME=HR1,RPGN=159
TRXNAME=HR2,RPGN=160
TRXNAME=DE1,RPGN=161
TRXNAME=SC2,RPGN=162
TRXNAME=SC4,RPGN=163
TRXNAME=SC6,RPGN=164
TRXNAME=TS1,RPGN=165
TRXNAME=OE1,RPGN=166
TRXNAME=OE2,RPGN=167
TRXNAME=OE4,RPGN=168
TRXNAME=OE5,RPGN=169
TRXNAME=IT1,RPGN=170
TRXNAME=IT2,RPGN=171
TRXNAME=IT8,RPGN=172
TRXNAME=PS2,RPGN=173
TRXNAME=PS3,RPGN=174
TRXNAME=PA2,RPGN=175
TRXNAME=DU1,RPGN=176
TRXNAME=DW1,RPGN=177
/*-----*/
TRXNAME=XFOR,RPGN=181
TRXNAME=YFOR,RPGN=182
TRXNAME>ZFOR,RPGN=183
TRXNAME=HX1,RPGN=184
TRXNAME=HX2,RPGN=185

```

```

TRXNAME=DX1,RPGN=186
TRXNAME=SX2,RPGN=187
TRXNAME=SX4,RPGN=188
TRXNAME=SX6,RPGN=189
TRXNAME=TX1,RPGN=190
TRXNAME=OX1,RPGN=191
TRXNAME=OX2,RPGN=192
TRXNAME=OX4,RPGN=193
TRXNAME=OX5,RPGN=194
TRXNAME=IX1,RPGN=195
TRXNAME=IX2,RPGN=196
TRXNAME=IX8,RPGN=197
TRXNAME=PX2,RPGN=198
TRXNAME=PX3,RPGN=199
TRXNAME=PF2,RPGN=200
TRXNAME=DY1,RPGN=201
TRXNAME=DZ1,RPGN=202
/*-----*/
SUBSYS=M1A2,RPGN=155          /* MRO PLEX #1 AOR #1          */
/*-----*/
TRXNAME=/FOR,RPGN=156
TRXNAME=UFOR,RPGN=157
TRXNAME=WFOR,RPGN=158
TRXNAME=HR1A,RPGN=108
TRXNAME=HX1A,RPGN=109
TRXNAME=HR2A,RPGN=110
TRXNAME=HX2A,RPGN=111
TRXNAME=DE1A,RPGN=112
TRXNAME=DX1A,RPGN=113
TRXNAME=SC2A,RPGN=114
TRXNAME=SX2A,RPGN=115
TRXNAME=SC4A,RPGN=116
TRXNAME=SX4A,RPGN=117
TRXNAME=SC6A,RPGN=118
TRXNAME=SX6A,RPGN=119
TRXNAME=TS1A,RPGN=120
TRXNAME=TX1A,RPGN=121
TRXNAME=OE1A,RPGN=122
TRXNAME=OX1A,RPGN=123
TRXNAME=OE2A,RPGN=124
TRXNAME=OX2A,RPGN=125
TRXNAME=OE4A,RPGN=126
TRXNAME=OX4A,RPGN=127
TRXNAME=OE5A,RPGN=128
TRXNAME=OX5A,RPGN=129
TRXNAME=IT1A,RPGN=130
TRXNAME=IX1A,RPGN=131
TRXNAME=IT2A,RPGN=132
TRXNAME=IX2A,RPGN=133
TRXNAME=IT8A,RPGN=134
TRXNAME=IX8A,RPGN=135
TRXNAME=PS2A,RPGN=136
TRXNAME=PX2A,RPGN=137
TRXNAME=PS3A,RPGN=138
TRXNAME=PX3A,RPGN=139
TRXNAME=DU1A,RPGN=140
TRXNAME=DW1A,RPGN=141
TRXNAME=DY1A,RPGN=142
TRXNAME=DZ1A,RPGN=143

```



```

TRXNAME=HR1,RPGN=159
TRXNAME=HR2,RPGN=160
TRXNAME=DE1,RPGN=161
TRXNAME=SC2,RPGN=162
TRXNAME=SC4,RPGN=163
TRXNAME=SC6,RPGN=164
TRXNAME=TS1,RPGN=165
TRXNAME=OE1,RPGN=166
TRXNAME=OE2,RPGN=167
TRXNAME=OE4,RPGN=168
TRXNAME=OE5,RPGN=169
TRXNAME=IT1,RPGN=170
TRXNAME=IT2,RPGN=171
TRXNAME=IT8,RPGN=172
TRXNAME=PS2,RPGN=173
TRXNAME=PS3,RPGN=174
TRXNAME=PA2,RPGN=175
TRXNAME=DU1,RPGN=176
TRXNAME=DW1,RPGN=177
/*-----*/
TRXNAME=XFOR,RPGN=181
TRXNAME=YFOR,RPGN=182
TRXNAME>ZFOR,RPGN=183
TRXNAME=HX1,RPGN=184
TRXNAME=HX2,RPGN=185
TRXNAME=DX1,RPGN=186
TRXNAME=SX2,RPGN=187
TRXNAME=SX4,RPGN=188
TRXNAME=SX6,RPGN=189
TRXNAME=TX1,RPGN=190
TRXNAME=OX1,RPGN=191
TRXNAME=OX2,RPGN=192
TRXNAME=OX4,RPGN=193
TRXNAME=OX5,RPGN=194
TRXNAME=IX1,RPGN=195
TRXNAME=IX2,RPGN=196
TRXNAME=IX8,RPGN=197
TRXNAME=PX2,RPGN=198
TRXNAME=PX3,RPGN=199
TRXNAME=PF2,RPGN=200
TRXNAME=DY1,RPGN=201
TRXNAME=DZ1,RPGN=202
/*-----*/
SUBSYS=M1F1,RPGN=205          /* MRO PLEX #1 FOR #1          */
/*-----*/
SUBSYS=IMS
SRVCLASS=IMS,RPGN=998
SRVCLASS=IMS1,RPGN=999
SUBSYS=CICS
SRVCLASS=CICS,RPGN=1000
SRVCLASS=CICUSRTX,RPGN=1001
SRVCLASS=CICSYSTX,RPGN=1002

```

Appendix C. IEAOPT Member used in the WLM Test Cases

The following is the OPT member used in the WLM test cases.

```
RCCUICT=(2,4),  
RCCCPUT=(101,106),  
SWAPRSF=10,  
MCCAECTH=(150,300),  
ESCTVIO=1500,  
ESCTBDS=1500,  
ESCTPOC(0)=1200,  
ESCTPOC(1)=1200,  
ESCTPOC(2)=1200,  
ESCTSWTC(0)=450,  
ESCTSWTC(1)=450,  
ESCTSWWS(0)=450,  
ESCTSWWS(1)=450,  
ESCTSWTC(2)=350,  
ESCTSWWS(2)=350,  
ESCTSTC(1)=250,  
ESCTSTC(2)=250,  
ESCTSTC(0)=100,  
ESCTVF=100
```

Appendix D. Service Definition — PERFPOL

The following is the definition of the policy used by the performance teams.

D.1 Overview

This policy had:

- 5 workloads
- 20 service classes
- 3 resource groups
- 13 service policies
- 0 classification groups
- 7 subsystem types
- 201 report classes

D.1.1 Service Coefficients

The service coefficients used to define the service units were:

- CPU 10.0
- IOC 5.0
- MSO 3.0000
- SRB 10.0

D.2 Classification Rules

The following were the classification rules used to establish the service classes for the various work.

Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	/FOR		CICUSRTX	FOR
2	SI	M1T1		CICSYSTX	FOR
1	TN	XFOR		CICUSRTX	XFOR
2	SI	M1T1		CICSYSTX	XFOR
1	TN	UFOR		CICUSRTX	UFOR
2	SI	M1T1		CICSYSTX	UFOR
1	TN	WFOR		CICUSRTX	WFOR
2	SI	M1T1		CICSYSTX	WFOR
1	TN	YFOR		CICUSRTX	YFOR
2	SI	M1T1		CICSYSTX	YFOR
1	TN	ZFOR		CICUSRTX	ZFOR
2	SI	M1T1		CICSYSTX	ZFOR
1	TN	HR1		CICUSRTX	HR1

Table 13 (Page 2 of 3). CICS Classification Rules. WLM-defined Subsystem

Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	HX1		CICUSRTX	HX1
1	TN	HR2		CICUSRTX	HR2
1	TN	HX2		CICUSRTX	HX2
1	TN	DE1		CICUSRTX	DE1
1	TN	DX1		CICUSRTX	DX1
1	TN	SC2		CICUSRTX	SC2
1	TN	SX2		CICUSRTX	SX2
1	TN	SC4		CICUSRTX	SC4
1	TN	SX4		CICUSRTX	SX4
1	TN	SC6		CICUSRTX	SC6
1	TN	SX6		CICUSRTX	SX6
1	TN	TS1		CICUSRTX	TS1
1	TN	TX1		CICUSRTX	TX1
1	TN	OE1		CICUSRTX	OE1
1	TN	OX1		CICUSRTX	OX1
1	TN	OE2		CICUSRTX	OE2
1	TN	OX2		CICUSRTX	OX2
1	TN	OE4		CICUSRTX	OE4
1	TN	OX4		CICUSRTX	OX4
1	TN	OE5		CICUSRTX	OE5
1	TN	OX5		CICUSRTX	OX5
1	TN	IT1		CICUSRTX	IT1
1	TN	IX1		CICUSRTX	IX1
1	TN	IT2		CICUSRTX	IT2
1	TN	IX2		CICUSRTX	IX2
1	TN	IT8		CICUSRTX	IT8
1	TN	IX8		CICUSRTX	IX8
1	TN	PS2		CICUSRTX	PS2
1	TN	PX2		CICUSRTX	PX2
1	TN	PS3		CICUSRTX	PS3
1	TN	PX3		CICUSRTX	PX3
1	TN	DU1		CICUSRTX	DU1
1	TN	DW1		CICUSRTX	DW1
1	TN	DY1		CICUSRTX	DY1
1	TN	DZ1		CICUSRTX	DZ1
1	TN	HR1A		CICUSRTX	HR1A
1	TN	HX1A		CICUSRTX	HX1A
1	TN	HR2A		CICUSRTX	HR2A
1	TN	HX2A		CICUSRTX	HX2A
1	TN	DE1A		CICUSRTX	DE1A
1	TN	DX1A		CICUSRTX	DX1A
1	TN	SC2A		CICUSRTX	SC2A
1	TN	SX2A		CICUSRTX	SX2A
1	TN	SC4A		CICUSRTX	SC4A
1	TN	SX4A		CICUSRTX	SX4A
1	TN	SC6A		CICUSRTX	SC6A
1	TN	SX6A		CICUSRTX	SX6A
1	TN	TS1A		CICUSRTX	TS1A
1	TN	OE1A		CICUSRTX	OE1A
1	TN	OX1A		CICUSRTX	OX1A
1	TN	OE2A		CICUSRTX	OE2A
1	TN	OX2A		CICUSRTX	OX2A
1	TN	OE4A		CICUSRTX	OE4A
1	TN	OX4A		CICUSRTX	OX4A
1	TN	OE5A		CICUSRTX	OE5A
1	TN	OX5A		CICUSRTX	OX5A
1	TN	IT1A		CICUSRTX	IT1A
1	TN	IX1A		CICUSRTX	IX1A

<i>Table 13 (Page 3 of 3). CICS Classification Rules. WLM-defined Subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	IT2A		CICUSRTX	IT2A
1	TN	IX2A		CICUSRTX	IX2A
1	TN	IT8A		CICUSRTX	IT8A
1	TN	IX8A		CICUSRTX	IX8A
1	TN	PS2A		CICUSRTX	PS2A
1	TN	PX2A		CICUSRTX	PX2A
1	TN	PS3A		CICUSRTX	PS3A
1	TN	PX3A		CICUSRTX	PX3A
1	TN	DU1A		CICUSRTX	DU1A
1	TN	DW1A		CICUSRTX	DW1A
1	TN	DY1A		CICUSRTX	DY1A
1	TN	DZ1A		CICUSRTX	DZ1A
1	TN	H*		CICUSRTX	HTRANS
1	TN	D*		CICUSRTX	DTRANS
1	TN	S*		CICUSRTX	STRANS
1	TN	T*		CICUSRTX	TTRANS
1	TN	O*		CICUSRTX	OTRANS
1	TN	I*		CICUSRTX	ITRANS
1	TN	P*		CICUSRTX	PTRANS
1	TN	CSPG		CICUSRTX	CSPG
2	SI	M1T1		CICSYSTX	CSPG
1	TN	CSPQ		CICUSRTX	CSPQ
2	SI	M1T1		CICSYSTX	CSPQ
1	TN	CSPS		CICUSRTX	CSPS
2	SI	M1T1		CICSYSTX	CSPS
1	TN	C*		CICUSRTX	CTRANS
2	SI	M1T1		CICSYSTX	CTRANS
		Default		CICS	CICSDEFT

<i>Table 14. IMS Classification Rules. IBM-defined Subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	HR*		IMS1	IMS1
		Default		IMS	IMS

<i>Table 15 (Page 1 of 2). JES Classification Rules. IBM-defined Subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	M1T1		CICSRGN	M1T1
1	TN	M1A1		CICSRGN	M1A1
1	TN	M1A2		CICSRGN	M1A2
1	TN	M1F1		CICSRGN	M1F1
1	TC	A		BATCHLOW	CLASSA
1	TC	B		BATCHHI	CLASSBC
1	TC	C		BATCHHI	CLASSBC
1	TN	CARP*			CARP
1	TN	JOB*		BATCHSPL	BATCHSPL
1	TN	COBB*			COBB
1	TN	T500*		BATCHTR	BATCHTR
1	TN	COB2*			COB2
1	TN	CPGM*			CPGM

<i>Table 15 (Page 2 of 2). JES Classification Rules. IBM-defined Subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	DB2J*			DB2J
1	TN	DPST*			DPST
1	TN	INUP*			INUP
1	TN	PAYR*			PAYR
1	TN	SALE*			SALE
1	TN	SLRP*			SLRP
1	TN	SORT*			SORT
1	TN	TIMS*			TIMS
1	TN	WHSE*			WHSE
1	TN	NPGM*			NPGM
		Default		DISC	DISC

<i>Table 16 (Page 1 of 2). STC Classification Rules. IBM-defined subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	IMS*		IMSCTL	IMS
1	TN	DSW*		IMSMP	DSW
1	TN	MPW*		IMSMP	MPW
1	TN	MPP*		IMSMP	MPP
1	TN	MPR*		IMSMP	MPR
1	TN	DBRC*		IMSMP	DBRC
1	TN	DLISAS*		IMSCTL	DLISAS
1	TN	DLIH*		IMSCTL	DLIH
1	TN	IRLM*			IRLM
1	TN	%MASTER*			MASTER
1	TN	ALLOCAS			ALLOCAS
1	TN	CATALOG			CATALOG
1	TN	CONSOLE			CONSOLE
1	TN	DUMPSRV			DUMPSRV
1	TN	GRS			GRS
1	TN	LLA			LLA
1	TN	PCAUTH			PCAUTH
1	TN	SMF			SMF
1	TN	TRACE			TRACE
1	TN	RASP			RASP
1	TN	VLF			VLF
1	TN	XCFAS			XCFAS
1	TN	IOSAS			IOSAS
1	TN	DLF			DLF
1	TN	VFETCH			VFETCH
1	TN	WLM			WLM
1	TN	ANTMAIN			ANTMAIN
1	TN	SMXC			SMXC
1	TN	SYSBMAS			SYSBMAS
1	TN	INIT			INIT
1	TN	JES2			JES2
1	TN	JESXCF			JESXCF
1	TN	JES3			JES3
1	TN	GTF			GTF
1	TN	SMS*			SMS
1	TN	RMF*			RMF
1	TN	COU*			COU
1	TN	VSR*			VSR
1	TN	RMFGAT			RMFGAT
1	TN	RMF*			RMF

<i>Table 16 (Page 2 of 2). STC Classification Rules. IBM-defined subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	TN	RT*			RTXA
1	TN	VTAM*			VTAM
1	TN	TPNS*			TPNS
1	TN	NET*			NET
1	TN	TCAS			TCAS
1	TN	SOF*			SOF
1	TN	JCOUNT*			JCOUNT
1	TN	JMON*			JMON
1	TN	PWTR*			PWTR
1	TN	APPC			APPC
1	TN	ASCH			ASCH
1	TN	OPC*			OPC
1	TN	RACF*			RACF
1	TN	XA5MSTR			XA5MSTR
1	TN	XA5DBM1			XA5DBM1
1	TN	IRLMWFI			IRLMWFI
		Default			STCDEFLT

<i>Table 17. TSO Classification Rules. IBM-defined Subsystem</i>					
Level	Qualifier Type	Qualifier Name	Starting Position	Service Class	Report Class
1	UI	JOB		TSOSPECL	TSOSPECL
1	UI	IBMUSER		TSOSPECL	TSOSPECL
1	UI	SEEWAH		TSOSPECL	TSOSPECL
1	UI	NADAS		TSOSPECL	TSOSPECL
1	UI	CHRIS		TSOSPECL	TSOSPECL
1	UI	DAVE		TSOSPECL	TSOSPECL
1	UI	MRG		TSOSPECL	TSOSPECL
1	UI	TRF		TSOSPECL	TSOSPECL
1	UI	FEP		TSOSPECL	TSOSPECL
1	UI	ALICE		TSOSPECL	TSOSPECL
1	UI	PHIL		TSOSPECL	TSOSPECL
		Default		TSO	TSO

D.3 Workload/Service Class Descriptions

Workload	Description
BATCH	Batch work
CICS	CICS work
IMS	IMS work
OTHER	Other work
TSO	TSO work

Service Class	Description
BATCHHI	Importance 5 batch for cbw2
BATCHLOW	Importance 5 batch for cbw2
BATCHSPL	Batch high priority
BATCHTR	Batch Thrashers
CICS	CICS transactions
CICSRGN	CICS regions vel70
CICSYSTX	CICS system transactions
CICUSRTX	CICS user transactions
DISC	Discretionary class
IMS	IMS transactions
IMSTL	IMS control regions vel70
IMSMP	IMS MPRs/MPPs vel50
IMS1	IMS hr* transactions
TSO	TSO transactions
TSOEVEN	TSO transactions
TSOODD	TSO transactions
TSOSPECL	Special TSO users
VEL50	Velocity 50
VEL70	Velocity 70
VEL80	Velocity 80

D.4 Report Classes

Report Class	Description
ALLOCAS	ALLOCAS ASID Report class
ANTMAIN	ANTMAIN ASID Report class
APPC	APPC ASID Report class
ASCH	ASCH ASID Report class
BATCHSPL	High priority batch
BATCHTR	Batch Thrashers
CARP	CBW2 CARP jobs
CATALOG	Catalog ASID Report class
CICSALL	All CICS xactions not in rpt cls
CICSDEFT	CICS Default report class
CLASSA	Report class for class A
CLASSBC	Report class for classes BC
COBB	CBW2 COBB jobs
COB2	CBW2 COB2 jobs
CONSOLE	Console ASID Report class
COU	COUNTS ASID Report class
COUNTS	COUNTS ASID Report class
CPGM	CBW2 CPGM jobs
CSPG	CICS CSPG Transactions
CSPQ	CICS CSPQ Transactions
CSPS	CICS CSPS Transactions
CTRANS	CICS Trans beginning with C

Report Class	Description
DBRC	DBRC Report class
DB2J	CBW2 DB2J jobs
DE1	CICS DE1 Transaction
DE1A	CICS DE1A Transactions
DISC	Discretionary Report Class
DLF	DLF ASID Report class
DLIH	DLIH Report class
DLISAS	DLISAS Report class
DPST	CBW2 DPST jobs
DSW	DSW Report class
DTRANS	CICS Trans beginning with D
DUMPSRV	DUMPSRV ASID Report class
DU1	CICS DU1 Transactions
DU1A	CICS DU1A Transactions
DW1	CICS DW1 Transactions
DW1A	CICS DW1A Transactions
DX1	CICS DX1 Transaction
DX1A	CICS DX1A Transactions
DY1	CICS DY1 Transactions
DY1A	CICS DY1A Transactions
DZ1	CICS DZ1 Transactions
DZ1A	CICS DZ1A Transactions
FOR	CICS FOR Transaction
GRS	GRS ASID Report class
GTF	GTF ASID Report class
HR1	CICS HR1 Transactions
HR1A	CICS HR1A Transactions
HR2	CICS HR2 Transaction
HR2A	CICS HR2A Transactions
HTRANS	cics trans beginning with H
HX1	CICS HX1 Transaction
HX1A	CICS HX1A Transactions
HX2	CICS HX2 Trasaction
HX2A	CICS HX2A Transactions
IMS	IMS Report class
IMSHR	IMS HR* Transactions
IMS1	IMS HR* Transactions
INIT	INIT ASID Report class
INUP	CBW2 INUP jobs
IOSAS	IOSAS ASID Report class
IRLM	IRLM Report class
IRLMWFI	IRLMWFI Report Class
ITRANS	CICS Trans beginning with I
IT1	CICS IT1 Transactions
IT1A	CICS IT1A Transactions
IT2	CICS IT2 Transactions
IT2A	CICS IT2A Transactions
IT8	CICS IT8 Transactions
IT8A	CICS IT8A Transactions
IX1	CICS IX1 Transactions
IX1A	CICS IX1A Transactions
IX2	CICS IX2 Transactions
IX2A	CICS IX2A Transactions
IX8	CIS IX8 Transactions
IX8A	CICS IX8A Transactions
JCOUNT	JCOUNT ASID Report class
JESXCF	JESXCF Report class
JES2	JES2 Report class
JES3	JES3 Report class
JMON	JMON ASID Report class
LLA	LLA ASID Report class

Report Class	Description
MASTER	Master ASID Report class
MPP	IMP MPP Report class
MPR	MPR Report class
MPW	MPW Report class
M1A1	CICS M1A1 Region
M1A2	CICS M1A2 Region
M1A3	CICS M1A3 Region
M1F1	CICS M1F1 Region
M1T1	CICS M1T1 Region
M2A1	CICS M2A1 Region
M2A2	CICS M2A2 Region
M2F1	CICS M2F1 Region
M2F2	CICS M2F2 Region
M2T1	CICS M2T1 Region
M3A1	CICS M3A1 Region
M3A2	CICS M3A2 Region
M3F1	CICS M3F1 Region
M3T1	CICS M3T1 Region
M4A1	CICS M4A1 Region
M4A2	CICS M4A2 Region
M4F1	CICS M4F1 Region
M4T1	CICS M4T1 Region
M5A1	CICS M5A1 Region
M5F1	CICS M5F1 Region
M5T1	CICS M5T1 Region
M6A1	CICS M6A1 Region
M6F1	CICS M6F1 Region
M6T1	CICS M6T1 Region
M7A1	CICS M7A1 Region
M7F1	CICS M7F1 Region
M7T1	CICS M7T1 Region
NET	NET ASID Report class
NPGM	CBW2 NPGM jobs
OE1	CICS OE1 Transactions
OE1A	CICS OE1A Transactions
OE2	CICS OE2 Transactions
OE2A	CICS OE2A Transactions
OE4	CICS OE4 Transactions
OE4A	CICS OE4A Transactions
OE5	CICS OE5 Transactions
OE5A	CICS OE5A Transactions
OPC	OPC ASID Report class
OTRANS	CICS Trans beginning with O
OX1	CICS OX1 Transactions
OX1A	CICS OX1A Transactions
OX2	CICS OX2 Transactions
OX2A	CICS OX2A Transactions
OX4	CICS OX4 Transactions
OX4A	CICS OX4A Transactions
OX5	CICS OX5 Transactions
OX5A	CICS OX5A Transactions
PAYR	CBW2 PAYR jobs
PCAUTH	PCAUTH ASID Report class
PS2	CICS PS2 Transactions
PS2A	CICS PS2A Transactions
PS3	CICS PS3 Transactions
PS3A	CICS PS3A Transactions
PTRANS	CICS Trans beginning with P
PWTR	PWTR ASID Report class(prtr sim)
PX2	CICS PX2 Transactions
PX2A	CICS PX2A Transactions

Report Class	Description
PX3	CICS PX3 Transactions
PX3A	CICS PX3A Transactions
RACF	RACF ASID Report class
RASP	RASP ASID Report class
RMF	RMF Report class
RMFGAT	RMFGAT ASID Report class(monIII)
RT	RTXA ASID Report class
RTXA	report class for rtxa
SALE	CBW2 SALE jobs
SC2	CICS SC2 Transaction
SC2A	CICS SC2A Transactions
SC4	CICS SC4 Transactions
SC4A	CICS SC4A Transactions
SC6	CICS SC6 Transactions
SC6A	CICS SC6A Transactions
SLRP	CBW2 SLRP jobs
SMF	SMF ASID Report class
SMS	SMS ASID Report class
SMXC	SMXC ASID Report class
SOF	SOF ASID Report class
SORT	CBW2 SORT jobs
STCDEFLT	Default report class for STC
STRANS	CICS Trans beginning with S
SX2	CICS SX2 Transactions
SX2A	CICS SX2A Transactions
SX4	CICS SX4 Transactions
SX4A	CICS SX4A Transactions
SX6	CICS SX6 Transactions
SX6A	CICS SX6A Transactions
SYSBMAS	SYSBMAS ASID Report class
SYSSTC	System Started Task report class
TCAS	TCAS ASID Report class
TIMS	CBW2 TIMS jobs
TPNS	TPNS ASID Report class
TRACE	TRACE ASID Report class
TRXCLASA	transaction class for batch
TRXCLSBC	transaction class for sort/batch
TSO	TSO Transactions
TSOSPECL	TSO Special Report Class
TS1	CICS TS1 Transactions
TS1A	CICS TS1A Transactions
TTRANS	CICS trans beginning with T
TX1	CICS TX1 Transactions
UFOR	CICS UFOR Transactions
VFETCH	VFETCH ASID Report class
VLF	VLF ASID Report class
VSR	VSR ASID Report class
VTAM	VTAM ASID Report class
WFOR	CICS WFOR Transactions
WHSE	CBW2 WHSE jobs
WLM	WLM ASID Report class
XA5DBM1	XA5DBM1 Report Class
XA5MSTR	XA5MSTR Report Class
XCFAS	XCFAS ASID Report class
XFOR	CICS XFOR Transactions
YFOR	CICS YFOR Transactions
ZFOR	CICS ZFOR Transactions

D.5 Service Definition Service Class Goals

Table 18. Service Definition Goals Sorted by Service Class

Service Class	Workload	Per	Duration	Imp	Goal
BATCHHI	BATCH	1		5	Velocity 16
BATCHLOW	BATCH	1		5	Velocity 2
BATCHSPL	BATCH	1		2	Velocity 80
BATCHTR	BATCH	1			Discretionary
CICS	CICS	1		2	Avg 00:00:00.300
CICSRGN	OTHER	1		2	Velocity 70
CICSYSTX	CICS	1		2	Avg 00:00:00.015
CICUSRTX	CICS	1		2	Avg 00:00:00.090
DISC	BATCH	1			Discretionary
IMS	IMS	1		2	Avg 00:00:00.170
IMSCTL	OTHER	1		2	Velocity 70
IMSMP	OTHER	1		2	Velocity 50
IMS1	IMS	1		2	Avg 00:00:00.084
TSO	TSO	1	2000	2	Avg 00:00:00.040
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
TSOEVEN	TSO	1	2000	2	Avg 00:00:00.040
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
TSOODD	TSO	1	2000	2	Avg 00:00:00.040
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
TSOSPECL	TSO	1		1	Velocity 99
VEL50	OTHER	1		2	Velocity 50
VEL70	OTHER	1		2	Velocity 70
VEL80	OTHER	1		2	Velocity 80

Table 19 (Page 1 of 2). Service Definition Goals Sorted by Workload

Workload	Service Class	Per	Duration	Imp	Goal
BATCH	BATCHHI	1		5	Velocity 16
	BATCHLOW	1		5	Velocity 2
	BATCHSPL	1		2	Velocity 80
	BATCHTR	1			Discretionary
	DISC	1			Discretionary
CICS	CICS	1		2	Avg 00:00:00.300
	CICSYSTX	1		2	Avg 00:00:00.015
	CICUSRTX	1		2	Avg 00:00:00.090
IMS	IMS	1		2	Avg 00:00:00.170
	IMS1	1		2	Avg 00:00:00.084
OTHER	CICSRGN	1		2	Velocity 70
	IMSCTL	1		2	Velocity 70
	IMSMP	1		2	Velocity 50
	VEL50	1		2	Velocity 50
	VEL70	1		2	Velocity 70
TSO	TSO	1	2000	2	Avg 00:00:00.040
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
	TSOEVEN	1	2000	2	Avg 00:00:00.040
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
		TSOODD	1	2000	2

<i>Table 19 (Page 2 of 2). Service Definition Goals Sorted by Workload</i>					
Workload	Service Class	Per	Duration	Imp	Goal
		2	2000	2	Avg 00:00:00.350
		3		5	Avg 00:00:02.500
	TSOSPECL	1		1	Velocity 99

Index

Special Characters

\$SRMSnnn 27

A

action trace 70
activate 3
address space monitoring 51
AOR 5, 100
APGRNG 9
APPC 103
AUX Paging Cost Rate 50
average response time 11, 82

B

batch 100
BATCH Workload
 Description 126
BATCHHI service class
 Classification Rules 123
 Description 126
 Service Definition 130
BATCHLOW service class
 Classification Rules 123
 Description 126
 Service Definition 130
BATCHSPL service class
 Classification Rules 123
 Description 126
 Service Definition 130
BATCHTR service class
 Classification Rules 123
 Description 126
 Service Definition 130
begin-to-end 20, 21
bottleneck 37
BTE 20, 21
buckets 12

C

cap slices 78
cap time slices 49
capping 16, 96, 103
Capping test case
 service distribution 97
 workload description 96
CARP report class
 Classification Rules 123
CICS 1, 91, 100
CICS service class
 Classification Rules 123
 Description 126

CICS service class (*continued*)
 Service Definition 130
CICS Workload
 Description 126
CICSplex/System Manager 3
CICSRGN service class
 Classification Rules 123
 Description 126
 Service Definition 130
CICSYSTX service class
 Classification Rules 121, 123
 Description 126
 Service Definition 130
CICUSRTX service class
 Classification Rules 121, 122, 123
 Description 126
 Service Definition 130
compatibility mode 1, 2, 7, 9, 59, 82, 99
couple data set 2
CP/SM 3
CPSM QUEUE algorithm 31
CPU delay 10
Customer Information Control System
 See CICS

D
DISC 27
DISC service class
 Classification Rules 124
 Description 126
 Service Definition 130
discretionary 6, 12
discretionary storage 51
dispatching priority 14, 71, 81
domain 81
donor 17, 18
donor selection 38
donors 34
DPRTY 14, 71
duration 5
dynamic internal server class 27

E

ESC 27
EXE 21
execution delay monitoring services 18, 20
execution phase 21
execution velocity 11, 61
external service class 27

F

fix routines 39
Fix_AUX_Storage_Delays Routine 39
FOR 100

G

goal mode 1, 2, 9, 59, 82, 103
goals 1, 4

H

HX1 report class
Classification Rules 122

I

I/O priority queueing 10, 14
ICS 2, 61
IEAICS 81
IEAICSxx 100
IEAIPS 81
IEAOPT 81
importance 4, 5, 6, 14, 91
IMS 1, 29, 100
IMS service class
Classification Rules 123
Description 126
Service Definition 130
IMS Workload
Description 126
IMS1 service class
Classification Rules 123
Description 126
Service Definition 130
IMSCTL service class
Classification Rules 124
Description 126
Service Definition 130
IMSMP service class
Classification Rules 124
Description 126
Service Definition 130
Information Management System
See IMS
internal service class names 68
Introduction to Workload Management 1
IPL 2
IPS 2
ISPF application 3

J

JES 100

L

local PI 11

M

mean-time-to-wait 14
migration 8, 82, 99
Mixed Workload test case
goals 91
performance index 94, 98
service distribution 93
workload description 91
MODE= 60
monitoring 51
monoplex 99
MPL delay 10
MPL_Delay_Fix Routine 43

N

Net Productive CPU Rate 51
net value calculation 35

O

OK1 action 55
OK1 Point 51
OTHER Workload
Description 126

P

PB 20
states 24
percentile 11
performance block
See PB
performance group 99
performance index 6
period summary 71
periods 4
pi 6
PI calculation 11, 13
PI calculation example 12
plots 41, 69
plotting 35
policy 2, 6
policy adjustment routine 10, 14, 15, 18, 34, 36, 44,
48, 49, 50
Process Storage OK1 Point 51
Processor Delay Fix 48
Productive CPU Rate 50

R

receiver 17
receiver selection 37
receivers 34
report performance group 100
reports in compatibility mode 60
reports in goal mode 61

- resource adjustment routine 10, 14, 16, 18, 50, 52, 53
- resource donor 17
- resource group 6, 103
- resource receiver 17
- response time 5
 - average 5
 - percentile 5
- response time distribution report 66
- response time goal 99
- response time percentile 11
- RMF 99
- RMF monitor I 59
- RMF monitor II 59
- RMF monitor III 13, 59
- RMF monitoring 3
- RMF report 33
- RMF reports 60, 61
- RMF Version 5.1 13
- RMF Workload Activity Report 82

S

- server topology 17
- server topology assessment 27
- service class 3, 4, 99
 - duration 5
 - importance 5
- service class names 68
- service definition 4, 121–131
 - Coefficients 121
 - Service Coefficients 121
- service level 82
- service level agreement 99
- service policy 6
- SMF 99 39, 67, 95
- SMF record 3
- SRM
 - algorithms 10
- srvclass 61, 100
- started tasks 101
- STOR delay 10
- Storage Stress test case
 - goals 87
 - performance index 89
 - RMF paging activity report 90
 - service distribution 88
 - workload description 87
- storage target 95
- storage targets 15
- subsystem work manager 6
- swap protect time 15
- Swap_Delay_Fix routine 46
- SWIN delay 10
- sysplex 99
- sysplex Performance Summary report 65
- sysplex PI 11
- system resource manager 2
 - See also* SRM

- system under-utilized 77

T

- test cases 81
- time slice function 14
- TOR 100
- Total Paging Cost Rate 50
- transaction routing 4
- TSO 81, 99
- TSO period 82
- TSO service class
 - Classification Rules 125
 - Description 126
 - Service Definition 130
- TSO test case
 - goals 82, 86
 - observations 85, 86
 - results 83, 86
 - RMF workload activity report 82
 - workload description 81, 85
- TSO Workload 81
 - Description 126
- TSEVEN Service Class
 - Description 126
 - Service Definition 130
- TSOODD Service Class
 - Description 126
 - Service Definition 130
- TSOSPECL service class
 - Classification Rules 125
 - Description 126
 - Service Definition 130

U

- Unmanaged CPU Paging Cost Rate 51

V

- VEL50 Service Class
 - Description 126
 - Service Definition 130
- VEL70 Service Class
 - Description 126
 - Service Definition 130
- VEL80 Service Class
 - Description 126
 - Service Definition 130
- velocity 5, 11
- velocity goal 99

W

- WLM
 - compatibility mode 1
 - couple data set 2
 - goal mode 1
 - goals 1

WLM (*continued*)
 introduction 1
 modes 1
 server topology 17
 terminology 4
 test cases 81
WLM services
 CICS Example 30
 classify 3, 19
 connect 19
 execution delay monitoring 20
 IMS example 29
 IWMCLSFY 20
 IWMCONN 19
 IWMDISC 20
 IWMCHST 24, 26
 IWMMCREA 25
 IWMMDELE 26
 IWMMINIT 25
 IWMMNTFY 26
 IWMMRELA 25
 IWMMSWCH 26
 IWMMXFER 24, 25, 26
 IWMPQRY 20
 IWMRPT 20
 IWRMCOLL 21
 report 19
work manager services 18, 19
Working Set Management 73
workload 81, 85
Workload Activity Report in Goal Mode 62
workload manager
 See WLM
workload manager delays report 66
workload manager modes 1
workload manager test cases 81
workload reporting services 18, 34

**International Technical Support Organization
System/390 MVS/ESA Version 5
Workload Manager Performance Studies
September 1995**

Publication No. SG24-4352-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

**Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

Overall Satisfaction	_____		
Organization of the book	_____	Grammar/punctuation/spelling	_____
Accuracy of the information	_____	Ease of reading and understanding	_____
Relevance of the information	_____	Ease of finding information	_____
Completeness of the information	_____	Level of technical detail	_____
Value of illustrations	_____	Print quality	_____

Please answer the following questions:

- a) If you are an employee of IBM or its subsidiaries:
- | | | |
|--|----------|---------|
| Do you provide billable services for 20% or more of your time? | Yes_____ | No_____ |
| Are you in a Services Organization? | Yes_____ | No_____ |
- b) Are you working in the USA? Yes_____ No_____
- c) Was the Bulletin published in time for your needs? Yes_____ No_____
- d) Did this Bulletin meet your needs? Yes_____ No_____

If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name

Address

Company or Organization

Phone No.



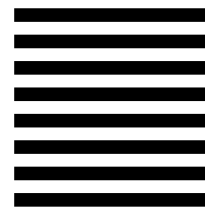
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Mail Station P099
522 SOUTH ROAD
POUGHKEEPSIE NY
USA 12601-5400



Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.

SG24-4352-00

