IBM

End to End Performance Management on IBM i

- Understand the cycle of Performance Management
- Maximize performance using the new graphical interface on V6.1
- Learn tips and best practices

Hernando Bedoya
Mark Roy
Nandoo Neerukonda
Petri Nuutinen

Redbooks

ibm.com/redbooks

International Technical Support Organization

End to End Peformance Management on IBM i

November 2009

SG24-7808-00

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (November 2009)

This edition applies to Version 6.1 of IBM i (5761-SS1).

© Copyright International Business Machines Corporation 2009. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices ix
Trademarks x

Preface xi
The team who wrote this book xi
Become a published author xiii
Comments welcome xiii

Part 1. Introduction to managing performance on IBM i 1

Chapter 1. Introduction to performance management 3
1.1 Why do performance management 4
1.2 Performance management made easy 4
1.3 What affects the performance 5
1.4 User expectations 5
1.4.1 Performance objectives 6
1.4.2 Performance measurement 7
1.4.3 Performance evaluation 7
1.5 Hardware performance 8
1.5.1 Central processing unit 8
1.5.2 Main storage 8
1.5.3 Disk 9
1.6 Work management setup 9
1.7 Software performance 9
1.7.1 IBM software performance 9
1.7.2 Database performance 9
1.7.3 Applications performance 10
1.8 Network performance 10
1.9 The need for a baseline 10
1.9.1 Some definitions 10

Chapter 2. Performance management life cycle 13
2.1 Describing the performance management life cycle 14
2.2 Planning for performance management 14
2.3 Performance management strategies 15
2.3.1 Performance management strategy for a small business 15
2.3.2 Performance management strategy for a mid-sized business 16
2.3.3 Performance management strategy for a large business 16
2.4 Create the baseline 16
2.4.1 Selecting data for a baseline 17
2.5 Monitor the partition performance 17
2.6 Collect the data 17
2.7 Analyze the data 18
2.8 Remove the identified performance problem 18
2.9 Tune the partition 18
2.10 Analyze the trend 19
2.11 Plan for capacity changes 20
2.12 Which tools to use during the cycle 20
2.12.1 IBM Systems Director Navigator for i performance interface 20

© Copyright IBM Corp. 2009. All rights reserved.

2.12.2 Collection Services . . . 21
2.12.3 IBM Performance Management for Power Systems . . . 21
2.12.4 System i Navigator Graph History function . . . 22
2.12.5 SQL performance Monitors . . . 22
2.12.6 Performance Tools for i . . . 23
2.12.7 Job Watcher . . . 23
2.12.8 Disk Watcher . . . 23
2.12.9 Performance Explorer . . . 24
2.12.10 Using the appropriate tool . . . 24

Part 2. IBM i tools . . . 27

Chapter 3. Overview of data collectors and tools . . . 29
3.1 Performance data collection overview . . . 30
3.1.1 Built-in metrics . . . 30
3.1.2 Wait accounting . . . 31
3.1.3 Storing collected data . . . 34
3.2 Performance data collectors overview . . . 34
3.2.1 Collection Services collector . . . 34
3.2.2 Job Watcher collector . . . 36
3.2.3 Disk Watcher collector . . . 36
3.2.4 Performance Explorer collector . . . 37
3.3 Data analysis and monitoring tools . . . 37
3.3.1 Relationship between collectors and tools . . . 38
3.3.2 Tool usage . . . 39
3.3.3 Tool packaging . . . 45

Chapter 4. IBM i Performance Data Collectors . . . 47
4.1 Introduction to data collectors . . . 48
4.2 Collection Services . . . 48
4.2.1 How Collection Services works . . . 48
4.2.2 Configuring Collection Services . . . 49
4.2.3 Customizing Collection Services . . . 54
4.2.4 Managing Collection Services data . . . 58
4.3 Job Watcher . . . 59
4.3.1 How Job Watcher works . . . 59
4.3.2 Configuring Job Watcher . . . 60
4.4 Disk Watcher . . . 62
4.4.1 How Disk Watcher works . . . 63
4.4.2 Configuring Disk Watcher . . . 63
4.5 Performance Explorer . . . 66
4.5.1 How Performance Explorer works . . . 67
4.5.2 Configuring Performance Explorer . . . 67
4.6 Managing collections . . . 70
4.6.1 Working with the Collections interface . . . 70
4.6.2 Working with collections . . . 74

Chapter 5. Performance data analysis . . . 81
5.1 Accessing the Investigate Data window . . . 82
5.2 Navigating around Investigate Data . . . 85
5.3 Investigating data . . . 91
5.3.1 Investigate using Health Indicators and Collection Services . . . 92
5.3.2 Using Job Watcher for further investigation . . . 108
5.3.3 Investigate using Disk Watcher data . . . 120

Chapter 6. Real-time monitoring tools 129
6.1 Management Central 130
6.2 Management Central monitors 130
6.2.1 Monitor types 130
6.2.2 Metrics 133
6.2.3 Thresholds 134
6.2.4 Configuring and starting a System Monitor 135
6.3 CL commands 143
6.3.1 Work with Active Jobs command 145
6.3.2 Work with System Status command 145
6.3.3 Work with Disk Status command 146
6.3.4 Work with System Activity command 147
6.4 IBM Systems Director Navigator for i 149
6.4.1 Performance tasks: real-time status 150
6.4.2 Health Indicators 154
6.5 Common Information Model 158
6.6 IBM Systems Director 158
6.6.1 IBM Systems Director components 159
6.6.2 Managing Power Systems 160
6.7 IBM Tivoli Monitoring 164
6.7.1 Monitoring Agent for IBM i 165
6.8 Watch for Event function 166
6.8.1 Watch for Event categories 166
6.8.2 Watch for event commands and APIs 167
6.8.3 Additional information 168

Chapter 7. Trending and capacity planning tools 169
7.1 The tools for trend analysis 170
7.1.1 Performance Management for Power Systems 170
7.1.2 Using Navigator for i for trending 177
7.1.3 Using CL commands for trend analysis 185
7.2 Tools for capacity planning 187
7.2.1 IBM Systems Workload Estimator (WLE) 187

Part 3. Best practices and appendixes 213

Chapter 8. Performance Management best practices 215
8.1 Back to the basics 216
8.1.1 System values 216
8.1.2 Work with System Status tips 218
8.1.3 Work with Disk Status tips 220
8.1.4 Work with Active Jobs tips 220
8.1.5 General tuning tips 221
8.2 Resource utilization guidelines 221
8.2.1 Memory 221
8.2.2 Disk storage 224
8.2.3 Disk response time 225
8.2.4 Disk percent busy 226
8.3 Key Performance Indicators 227
8.3.1 Hardware resource utilization 227
8.3.2 Throughput 228
8.4 Conducting a capacity planning cycle 228
8.4.1 Review current performance 228
8.4.2 Select representative data 228

8.4.3 Estimation 229
8.5 Effect of processor changes 230
8.6 Tools for testing application performance 230
8.7 Retaining collected data 231
8.7.1 Store data centrally 231
8.7.2 Keep some data online 231
8.7.3 Saving data to offline media 231
8.8 Collecting Collection Services data for an entire server 232
8.9 Disaster recovery and other failover systems 234
8.10 Web application performance 235
8.10.1 Web Performance Advisor 235
8.10.2 Web Performance Monitor 237
8.11 Performance Capabilities Reference 239

Appendix A. Understanding disk performance metrics 241
Part 1: Why disk performance is important 242
Why disk performance 242
Wait state accounting 242
Analyzing disk I/O time for a job 243
Transactional jobs considerations. 245
Where to go from here 246
Part 2: Introduction to disk operation on IBM i 246
Basic flow 247
Storage adapter caches 247
Read cache. 248
Write cache 249
Impact of different disk protection levels. 249
Parallel disk operations. 250
External disks versus internal disks 251
Multi-path support 252
IOPless storage adapters 252
Part 3: Metrics of disk performance 253
Disk performance data 253
Metrics of disk performance 254
Disk response time 255
Disk response time buckets 256
Disk percent busy 257
Disk service time and disk wait time 258
Read cache hit ratio 259
Write cache overrun ratio 259
Total disk device operation rate 260
Write cache efficiency 261
Operation rate for read/write operations 261
Average operation size 262
The queries 262

Appendix B. A partition as a business 267
Explaining your partition as a business 268
What the subsystem looks like. 268
Where the work comes from 268
Where the jobs find the space 268
How the work comes in 269
How the work is treated 269

Related publications 271
IBM Redbooks 271
Online resources 271
How to get Redbooks 271
Help from IBM 272

Index 273

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

© Copyright IBM Corp. 2009. All rights reserved.

Trademarks

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

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

Active Memory™	iSeries®	Redbooks®
AIX®	Lotus®	Redpapers™
AS/400®	OpenPower®	Redbooks (logo) ®
BladeCenter®	OS/400®	System i®
DB2®	Power Systems™	System p®
Domino®	POWER5™	System x®
eServer™	POWER6®	System z®
i5/OS®	POWER®	Tivoli®
IBM®	Rational®	WebSphere®

The following terms are trademarks of other companies:

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

Oracle, JD Edwards, PeopleSoft, Siebel, and TopLink are registered trademarks of Oracle Corporation and/or its affiliates.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

Monitoring and managing your system's performance is critical to ensure that you are keeping pace with the changing demands of your business.

To respond to business changes effectively, your system must change too. Managing your system, at first glance, might seem like just another time-consuming job. But the investment soon pays off because the system runs more efficiently, and this is reflected in your business. It is efficient because changes are planned and managed.

Managing performance of any system can be a complex task that requires a thorough understanding of that system's hardware and software. IBM® i is an industry leader in the area of performance management and has many qualities that are not found in other systems, such as:

- Unparalleled performance metrics
- Always-on collection of metrics
- Graphical investigation of performance data

While understanding all the different processes that affect system performance can be challenging and resolving performance problems requires the effective use of a large suite of tools, the functions offered by IBM i are intended to make this job easier for users.

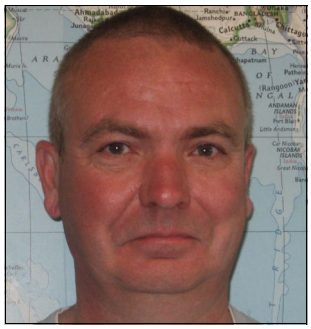
This IBM Redbooks® publication explains the tasks and rich tools associated with performance management on IBM i.

The team who wrote this book

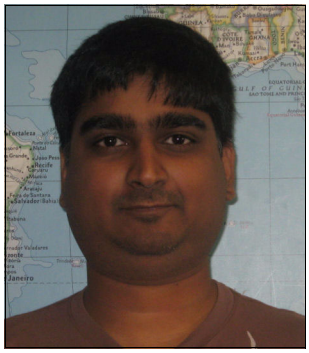
This book was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center.

Hernando Bedoya is a Senior IT Specialist at STG Lab Services and Training, in Rochester, Minnesota. He writes extensively and teaches IBM classes worldwide in all areas of DB2® for i. Before joining STG Lab Services, he worked at the ITSO for 9 years and wrote multiple IBM Redbooks publications. He also worked for IBM Colombia as an IBM AS/400® IT Specialist doing presales support for the Andean countries. He has 25 years of experience in the computing field and has taught database classes in Colombian universities. He holds a Master degree in Computer Science from EAFIT, Colombia. His areas of expertise are database technology, performance, and data warehousing. Hernando can be contacted at hbedoya@us.ibm.com.

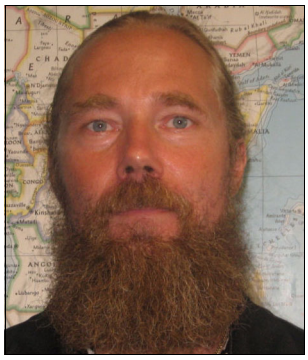
© Copyright IBM Corp. 2009. All rights reserved.



Mark Roy is an IBM i specialist based in Melbourne, Australia. He was previously with a large Australian bank, designing, installing, and enhancing their IBM i (previously branded as CPF, OS/400®, and i5/OS®) environment. During that time, he was responsible for the automation and global consolidation of their infrastructure, implementing robust disaster recovery and high availability solutions to support the centralized environment. Mark has previously authored two IBM Redbooks publications, *IBM i 6.1 Technical Overview*, SG24-7713 and *i5/OS Diagnostic Tools for System Administrators: An A to Z Reference for Problem Determination*, SG24-8253. He was also a member of the IBM Rochester's Performance Advisory Council. He recently established Sysarb, a company providing freelance consulting and contracting services to IBM i customers and service providers. He specializes in IBM i infrastructure architecture, technical support, and performance and systems management. Mark can be contacted at Mark.Roy@sysarb.com.au.



Nandoo Neerukonda is an IBM i consultant specializing in performance management, query optimization on DB2 for i, high availability, systems programming, and security. He has 14 years of experience with IBM i and its predecessors and has worked for Countrywide Financial (now Bank Of America) and Penske Truck Leasing. He currently provides consulting services through his own corporation, Metixis, Inc. He is a speaker at COMMON and a co-author of another IBM Redbooks publication, *DB2 Univeral Database for iSeries Administration: The Graphical Way on V5R3*, SG24-6092. Nandoo can be contacted at nandoo.neerukonda@metixis.com.



Petri Nuutinen is an IT specialist at IBM Finland. He started working with s/38 in 1982, and in 1987 he joined IBM and was responsible for the technical support of the AS/400 MRI translation process. His technical specialities are work management and performance management, and he has been a co-author for other IBM System i® Redbooks publications. He has participated in several AS/400, IBM eServer™ iSeries® (and all the other names for the system currently known as IBM System i) performance-related residencies and book projects since 1991. Petri currently divides his working time between pre- and postsales activities and billable services. The majority of his duties are related to managing a farm of IBM i partitions that are used to provide On Demand processing facilities for the customers of the outsourcing business where he manages the whole cycle of system performance management, from planning for the hardware and software requirements together with the customer to the end of the LPAR's life. Petri can be contacted at petri.nuutinen@pp.inet.fi.

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

Al Schwab
Mike Schwartz
Ann Lund
Pete Bertolozzi
Linda Robinson
International Technical Support Organization, Poughkeepsie Center

Helen Olson-Williams
Ronald Forman
Tim Rowe
Dave Legler
Alexei Pytel
Dawn May
Adam T Clark
Pete Heyrman
Ron McCargar
Lora Powell
Ron Fletcher
Stacy Benfield
Doyle Mccoy
Eric Barsness
IBM Development Rochester, MN

Lloyd Perera
IBM Performance Consultant

Become a published author

Join us for a two- to six-week residency program! Help write a book dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You will have the opportunity to team with IBM technical professionals, Business Partners, and Clients.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you will develop a network of contacts in IBM development labs, and increase your productivity and marketability.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Introduction to managing performance on IBM i

In this part, we introduce the terms and the basic methods of successfully managing performance on your IBM i partition(s). This part includes the following chapters:

- Chapter 1, “Introduction to performance management” on page 3 covers the factors of which the partition performance consists.
- Chapter 2, “Performance management life cycle” on page 13 describes the never ending process of managing the partition performance and suggests which of the tools should be used during the steps of this process.

© Copyright IBM Corp. 2009. All rights reserved.

1

Introduction to performance management

This chapter provides a short introduction to all aspects of performance management.

Monitoring and managing your partition's performance is critical to ensure you are keeping pace with the changing demands of your business. To respond to business changes effectively, your partition must change too.

Managing the performance of any partition can be a complex task that requires a thorough understanding of that partition's hardware and software. The IBM i V6.1 operating system is an industry leader in the area of performance management and has many qualities that are not found in other computer systems, including unparalleled performance metrics, data collection services that are constantly active, and graphical viewing of performance data. Understanding all the different processes that affect partition performance can be a challenging task and resolving performance problems may require effective use of a large suite of tools. The functions offered by the IBM i V6.1 operating system are intended to make this job easier for you.

Performance can be measured, monitored, and analyzed with various tools. In this book, we show the IBM provided tools available with the IBM i V6.1 operating system.

We use the term "partition" to specify a computing environment running on one copy of the operating system, whether by using all the resources in the physical system or by being a logical partition.

© Copyright IBM Corp. 2009. All rights reserved.

1.1 Why do performance management

Successfully managing performance ensures that your partition is efficiently using its resources and that your server provides the best possible services to your users and to your business needs.

Performance management is essential to optimizing the utilization of your computer system by measuring current capabilities, recognizing trends, and making appropriate adjustments to satisfy user and management requirements, such as response time or job throughput. It is needed to maintain business efficiency and avoid prolonged suspension of normal business activities. Managing the performance should be a part of your daily operations.

Understanding the factors that affect partition performance helps you respond to problems and make better long-term plans. Effective planning can prevent potential performance problems from developing and ensures that you have the system capacity to handle your current and growing workloads.

Lack of performance management, especially when a new release is installed, when a new application is added, or the workload has increased can lead to a severe business impact.

1.2 Performance management made easy

As the usage of information technology services change and functionality evolves, the usage of partition hardware resources also changes. Understanding the current workload processing demands and how they will change over time makes for easier and proactive planning for IT service growth.

If there are spikes in, for example, processing power used at a particular time of the day, you should analyze what is happening at that time. Then you may make changes to maximize the existing infrastructure, for example, tune the application, or move batch jobs to a quieter period of time.

Performance management activities are intended to optimize performance and efficiency, and to plan for and justify financial investments. Performance management is an ongoing process consisting of:

- Setting the performance objectives and creating the baseline
- Collecting the performance data on a partition
- Analyzing the collected data
- Adjusting the system parameters, if need be
- Analyzing the performance trend
- Capacity planning

This may also be seen as a flowchart, as shown in Figure 1-1.

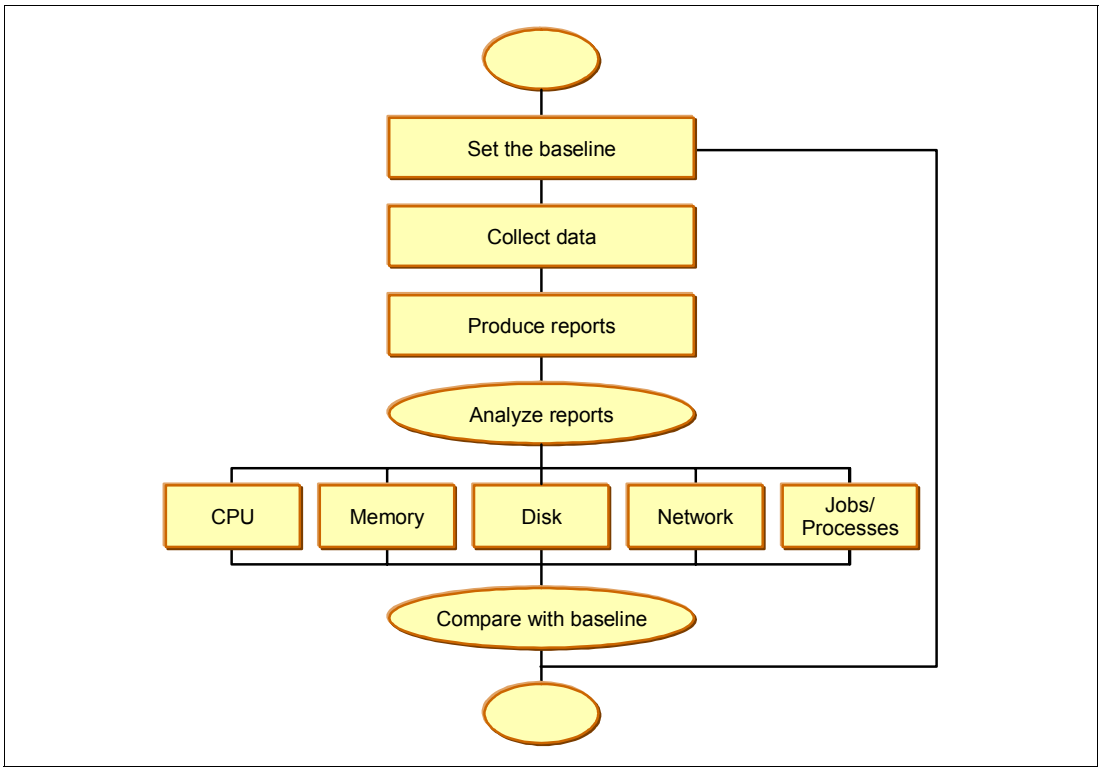


Figure 1-1 The performance management flowchart

1.3 What affects the performance

The factors affecting the performance are:

- User expectations
- Hardware capabilities and configuration
- Software, such as licensed programs, applications, and so on
- Workload, that is, the size of it and the rate with which jobs are being run
- Network

1.4 User expectations

The users on a system have expectations on the performance level of the computing environment on which they are running. Users expect the business functions to be available on time and stay available. Users expect that all the requests they enter are fulfilled within a normal time frame. Company management expects the IT department to provide a stable support for the business functions while not spending too much money for the costs. These all are important factors, but you must be able to measure them and express the measurements in an organized way.

You need to specify the performance objectives in a manner that can be measured. This includes specifications for batch and interactive work, such as:

- Average response time for the interactive transactions
- Number of batch jobs run daily and the average duration of those jobs
- Expected amount of interactive transactions per hour and per day

Remember that both predictability and timeliness of the run time of a request is what performance is all about.

1.4.1 Performance objectives

You must define the performance objectives for a partition as a part of the performance management strategy. These can be something simple such as having 75 percent of transactions completing in a time of less than two seconds.

The definition of batch job throughput can be based on a time period and records processed or lines of printed output produced during that time period. You should be aware of what can and cannot be expected from the current partition configuration. If a purchased application package is used, the performance information for a similar configuration should be reviewed. For example, the performance of an application with four active workstations should not be used to indicate performance when running the same application in an environment of 500 active workstations.

Response time

The definition of a response time is the elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response. The length of the time between pressing Enter on a submit command screen and the display of the first character of the response at a user terminal is an example of response time. This also applies to the multi-tiered applications where the transaction may have to perform processing on several servers before the request is fulfilled.

There is also the concept of a perceived response time, which is the time a user sees as the beginning of input and the end of the response.

Throughput

The definition of a throughput is the amount of work performed by a computer within a given time. It is a combination of internal processing speed, I/O functions, and the efficiency of the operating system, other system software, and applications all working together. Transactions processed per second is one metric commonly used to measure the throughput. Please remember that throughput includes all the work done within the partition.

System availability

Users expect the system to be available according to the Service Level Agreement (SLA). Availability is one component of performance management that must not be forgotten. The goal of performance management is to make sure that the partition is available within the SLA parameters. This is achieved by performing trend analysis and taking corrective actions as needed.

You cannot maintain system availability without having at least some kind of performance management strategy in place.

1.4.2 Performance measurement

This step covers the methodology for planning and executing performance measurements if you do not collect the data on a 24x7 basis. Performance measurement criteria should be formalized in order to perform a periodic run of the data collection. This criteria should define such things as:

- Day and time to collect data
- Duration of the collection
- Type of information to be collected

You should define a peak period of time (or critical applications) to be measured. Performance data can be collected from:

- Collection Services, which is the primary tool used to collect system performance data on IBM i.
- Observations made by the system operator or users at their workstations.

Using both sets of data is a good way to evaluate system and application performance. Observations made by operators or users should include the date and time of day, user ID, application function, or display function being performed and other information that may be unique to a particular application environment. If a problem occurs, any unusual findings should be documented. However, this kind of observation may not give detailed information about all components or jobs that might be needed for a problem analysis.

It is also important to collect performance data and review it even when there have been no reported performance problems so that potential problems may be detected before they impact the daily production environment.

1.4.3 Performance evaluation

You should evaluate the partition's performance on a regular basis, whether it is once a week or once a month. Compare the current measurements with the baseline and take corrective actions.

Figure 1-2 shows the components that make up the response time for an interactive transaction.

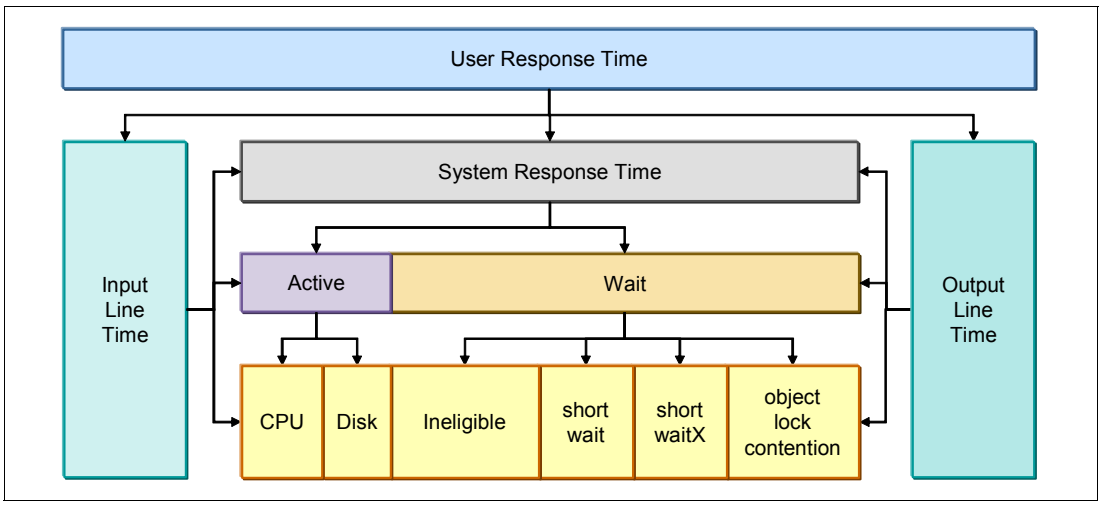


Figure 1-2 Interactive response time components

The components affecting the response time are:

Input Line Time	Network delay before reaching the system.
CPU	Central Processing Unit cycles used.
Disk	Disk I/O time.
Ineligible	Waiting for an activity level.
Short Wait	Waiting at an activity level.
Short Wait Extended	Not holding an activity level.
Object Lock Contention	Another job is holding the object.
Output Line Time	Network delay.
User Response Time	The processing time plus the network delay.
System Response Time	The partition internal response time, without network delay.

1.5 Hardware performance

Once performance objectives are set, the next task is to determine the hardware that can handle your workload and meet the objectives. To assess your current hardware capabilities, you need to determine average disk use, average and regular peaks of CPU use, and average memory use, among other parameters within the performance investigation.

Three key hardware-related components of a balanced partition are:

- CPU
- Memory
- Disk

You need a well balanced amount of each resource. Too large a portion of any one of these resources is not cost effective, as it does not increase the processing capability of the partition. An undersized component limits the processing capability of the partition.

1.5.1 Central processing unit

The central processing unit (CPU) is the part of the computer that actually does the processing of the data. Normally, the CPU should not utilize more than 75 percent of its capacity, as there should be some room for a temporary increase of the workload. You will start experiencing fluctuating response times and a decrease in throughput if the CPU load constantly reaches 100%.

1.5.2 Main storage

The data must be brought into the main storage before it can be processed with the CPU. The use of the main storage can only be measured indirectly, by measuring the rate of moving the data between the main storage and disk subsystem.

1.5.3 Disk

A disk is where data is permanently stored. The main aspects of measuring disk related data are:

- How much of the available disk space is being used
- How busy the disk arms are

Disk service time, disk seek time, and disk latency are usually investigated as a part of a detailed performance problem analysis. IBM i Disk Watcher can analyze the disk subsystem performance, as discussed in 4.4.2, “Configuring Disk Watcher” on page 63. Appendix A, “Understanding disk performance metrics” on page 241 has detailed information about disk performance metrics.

1.6 Work management setup

A work management setup that is appropriate for all types of a partition does not exist. The shipped setup will get your partition configured and running, but to get the most out of your hardware investment, you need to properly tune your partition and keep it in tune as your business evolves. As the work management setup on your partition affects the performance, you should follow one principle: favor output over input.

See Appendix B, “A partition as a business” on page 267 for one explanation of work management.

1.7 Software performance

Software performance can and should be defined, divided, and investigated into the areas of IBM software performance, database performance, and applications performance.

1.7.1 IBM software performance

You should verify that you are running the latest functioning level of code and have the most up to date Fix Packs and Program Temporary Fixes. The easiest way to update your code is to go to the following address:

http://www-912.ibm.com/s_dir/slkbase.nsf/recommendedfixes

Compare the information at this address with the applied PTF levels on your partition and update if necessary.

1.7.2 Database performance

Database performance depends on the indexing strategy, the database structure, and other factors. These can be analyzed with SQL Performance Monitors within the IBM Navigator for i. However, database performance is also dependent on the overall hardware configuration and performance of a partition, especially the disk subsystem and both the amount of memory and the memory pool setup.

1.7.3 Applications performance

Collection Services is a good tool to start your investigation about an application's workload.

Investigating the application's performance depends on how monitoring the application performance is instrumented within the application. It might be a good idea to prepare an application with hooks to enable performance statistics collection. This procedure is described in the Information Center at the following address:

http://publib.boulder.ibm.com/infocenter/iseries/v6r1m0/index.jsp

Select **Systems management** → **Performance** → **Applications for performance management** → **Collection Services** → **Implementing user-defined categories in Collection Services** to access this information.

1.8 Network performance

The network is one of the contributors to the overall performance of a given partition. Sometimes problems in the network may appear to a user to be a performance problem on a given partition, but a quick investigation usually determines that it is the network that is experiencing the problem.

1.9 The need for a baseline

Setting up a baseline may be easy, as the system has to be able to process the daily workload before the next working day starts. When the size of the daily workload is known, it is relatively simple to correlate that workload with the system resource utilization.

Whether you use a simple or a complex baseline, you need one or you will not be able to detect how your partition performance changes. This is especially true if you are responsible for handling the performance of multiple partitions.

You must define a baseline for system performance before you can successfully follow any of the strategies described in Chapter 2, "Performance management life cycle" on page 13.

1.9.1 Some definitions

The list of items discussed in this section is not a complete list of performance related factors, but describes some of the parts of performance.

Interactive work

In general, interactive workloads all have the characteristics of a 5250 workstation communications path somewhere within the workload. Many older hardware models allowed selecting different interactive features depending upon interactive processing needs. The interactive feature was the feature code that represented the total interactive processing capacity of the particular system. The IBM Systems Workload Estimator (WLE) tool will check interactive needs against the interactive features and select systems with sufficient interactive capacities when sizing against older hardware. WLE is discussed in 7.2.1, "IBM Systems Workload Estimator (WLE)" on page 187.

Batch work

Non-interactive workloads are those not requiring a 5250 workstation communications path. These are typically client/server applications (for example, WebSphere® Commerce Suite, Domino®, HTTP, FTP, and so on) or batch/background applications.

Commercial Processing Workload (CPW)

CPW stands for Commercial Processing Workload, which should be contrasted with Computational Intensive Workload (CIW). Commercial workloads are characterized by a relatively large amount of I/O with respect to computation. These values should *not* be used as a System Processor Capacity due to the nature of the workload. Appendix A, "Understanding disk performance metrics" on page 241 contains a complete explanation of CPW and CIW. Also, remember that these values are only relative to other Power System, IBM System i, IBM eServer iSeries, or AS/400 system models when the workloads are of a commercial nature.

CIW

These workloads are characterized with a relatively large amount of computations with respect to the amount of I/Os performed. Workloads that can be characterized as computational intensive include Domino Mail, some Java™ applications, and so on.

Disk arms

Modern disk drives have only one disk read/write arm, so the number of disk arms is the same as the number of disk drives. In general, the more disk arms/drives a system has, the more quickly it can access information and the better the performance.

Disk capacity

Disk capacity is the amount of disk storage space, typically given in GB. In general, larger workloads or databases or larger numbers of users typically require more storage space. However, there are some applications that need plenty of disk arms for the proper functionality.

Memory paging and faulting

As all the data residing in disk does not usually fit into the main storage at the same time, it is brought into memory in chunks as needed.

Paging

Paging is the movement of data in and out of memory, both synchronously and asynchronously. Pages can be written out to storage or removed from memory without being written if they have not been changed. Faulting causes paging to occur on the server.

Faulting

Faulting occurs when a referenced page, or piece of data, is not in memory. This causes programs to stop because they must wait for the data to be paged in. Faulting is normal behavior on IBM i. However, the size of the storage pools can affect the faulting rates, and storage pools that are too small can result in excessive faulting.

You can examine and set the storage pool sizes by using the Work with System Status (WRKSYSSTS) command. If you have too many active threads or threads that use too many system resources, you will see numerous page faults for the storage pools in which applications are running. Increasing the storage pool size or decreasing the activity level might reduce the number of page faults.

2

Performance management life cycle

The Performance Management process is a cycle of tasks consisting of thorough planning, setting the baseline, monitoring the system, and collecting the performance data. Data is analyzed, first to decide if there is a performance problem, and then by drilling down to discover and remove the root cause of the problem. The next step is to perform the historical trending and, if needed, perform the capacity planning to discover the changes needed in the hardware or software.

© Copyright IBM Corp. 2009. All rights reserved.

2.1 Describing the performance management life cycle

The performance management life cycle starts with outlining the business requirements and completes when the first set of performance reports are analyzed and adjustments are performed.

The cycle can be seen as a flowchart, as shown in Figure 2-1.

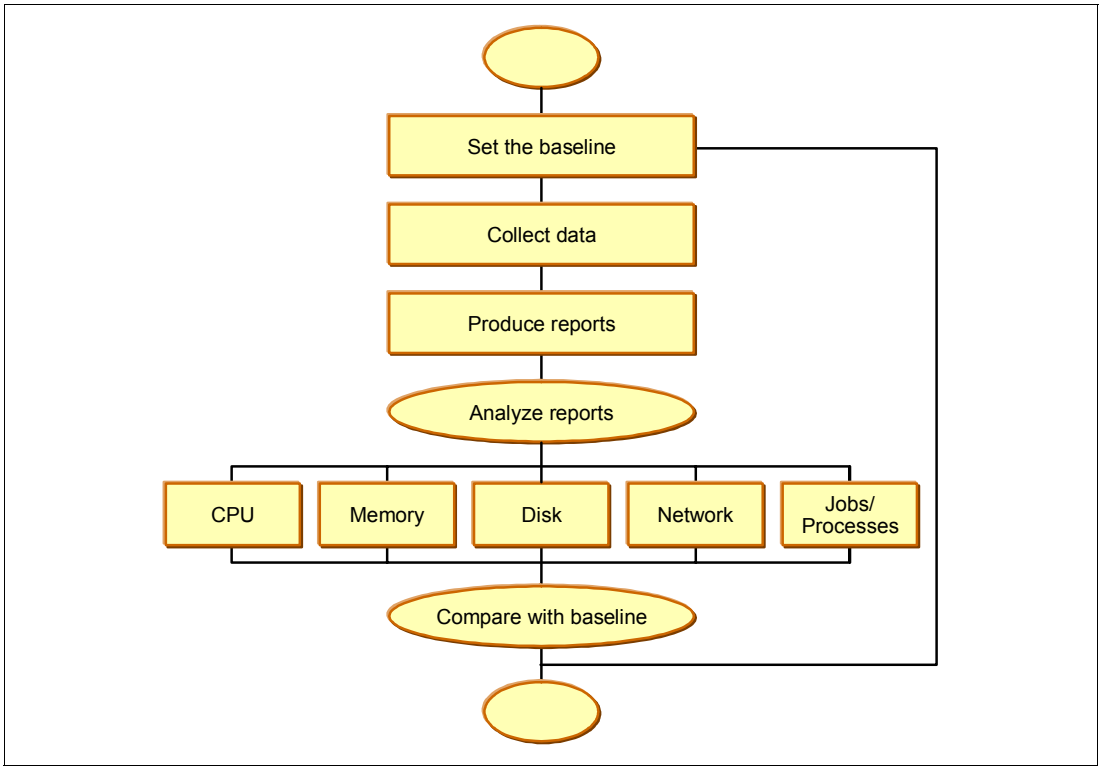


Figure 2-1 Performance management life cycle as a flowchart

2.2 Planning for performance management

The first task in a system performance or capacity study is to make sure that all the involved parties reach a clear understanding of the performance goals (as measured by the users). The partition performance criteria should be clearly stated and be understood before doing the analysis and especially before making any changes. You may have a mix of applications with performance criteria that vary over time, depending upon the currently active applications. When interactive and non-interactive jobs run concurrently, they may have conflicting performance requirements.

Both interactive and non-interactive processing objectives should be defined in terms that can be measured either automatically by the system or through monitoring by a person. An objective must also be reasonable based on the complexity of the processing and the power of the system model being used.

Start the performance definition work by establishing specific performance criteria and system resource utilization guidelines to be used to evaluate system performance. The user may require that a certain set of non-interactive jobs must run every day at the close of business and the jobs must complete the run in one hour. That is a good objective from the user's point of view, but there are other things to consider.

Remember that the objective of a performance definition effort is to translate the user's system performance needs into a well-defined statement of goals and requirements.

2.3 Performance management strategies

Developing a good performance management strategy will help you manage your partition's performance proactively instead of reacting to changes that have already taken place.

Your performance management strategy depends in a large part on how much time you can afford to spend managing performance. If you are working with a small company, you may be managing many different aspects of your business and cannot dedicate many hours to managing performance. Many large companies employ performance specialists to keep their systems tuned and running effectively.

Different business needs require different performance management strategies. For determining a basic performance management strategy and for identifying which performance applications to use, classify your company in one of three categories:

- Small business
- Mid-sized business
- Large business

The business resources vary for each size, and your management strategy will vary accordingly.

2.3.1 Performance management strategy for a small business

A small business most likely has fewer resources to devote to managing performance than a larger business. For that reason, use as much automation as possible. You can use the IBM Performance Management (PM) for Power Systems™ service offering to have your performance data sent to IBM where it will be compiled and generated into reports for your use. This not only saves you time, but IBM also makes suggestions to you when your server needs an upgrade.

The following is a list of recommended performance applications for a small business:

- IBM Systems Director Navigator for i Performance interface to manage and investigate.
- Collection Services to collect sample data at user defined intervals for later analysis.
- Performance Management for Power Systems to automate the collection, archival, and analysis of system performance data.
- System i Navigator Monitors to observe graphical representations of system performance, and automate responses to predefined events or conditions.

2.3.2 Performance management strategy for a mid-sized business

A mid-sized business probably has more resources devoted to managing performance than a small one. You may still want to automate as much as possible and can also benefit from using PM for Power Systems.

The following is a list of recommended performance applications for a mid-sized business:

- IBM Systems Director Navigator for i Performance interface to display and manage performance data.
- Collection Services to collect sample data at user-defined intervals for later analysis.
- Performance Management for Power Systems to automate the collection, archival, and analysis of system performance data.
- Performance Tools for i to gather, analyze, and maintain system performance information.
- System i Navigator Monitors to observe graphical representations of system performance, and automate responses to predefined events or conditions.
- Job Watcher to collect detailed information about a specific job or thread resource use.

2.3.3 Performance management strategy for a large business

The following is a list of recommended performance applications for a large business:

- IBM Systems Director Navigator for i Performance interface to display and manage performance data.
- Collection Services to collect sample data at user-defined intervals for later analysis.
- Performance Management for Power Systems to automate the collection, archival, and analysis of system performance data.
- Performance Tools for i to gather, analyze, and maintain system performance information.
- System i Navigator Monitors to observe graphical representations of system performance, and automate responses to predefined events or conditions.
- Job Watcher to collect detailed information about a specific job or thread resource use.
- Disk Watcher to collect detailed information about disk performance data.
- Performance Explorer (PEX) to collect detailed information about a specific application or system resource.
- IBM Tivoli® monitoring may be used as a monitoring solution.
- IBM Systems Director for monitoring.

2.4 Create the baseline

A baseline is a self-made tool, something like a ruler with the exception that you cannot buy a baseline from any shop. You are the one responsible for creating the baseline for your use.

You need a baseline to spot the changes in the partition performance.

You compare the current partition performance measurements with the baseline. The baseline should consist of multiple aspects of the partition usage and it can be something simple such as:

- Average response time of 0.75 seconds.
- End-of-day batch jobs must complete in four and half hours.
- Average CPU utilization during business hours is below 85 per cent.
- Disk used percentage is below 70 per cent.
- Number of interactive transactions per day is 20,000.

The baseline may also be something more complicated. For example, in a multi-tiered application, it may be useful to create a benchmark transaction or a chain of transactions that tests all the aspects of the application structure.

One easy method of creating your baseline is using Collection Services, because it is running all the time.

2.4.1 Selecting data for a baseline

Baselines are usually captured for one of the following reasons:

- As a reference point to measure improvements when resolving performance issues.
- As a reference point before a major system change, such as:
  - Application
  - Operating system
  - Hardware
- As a reference point for a capacity planning exercise.

Where possible, baselines should be collected on an unconstrained system, particularly when used as a baseline for capacity planning purposes.

2.5 Monitor the partition performance

Once the baseline is created, you need to monitor the performance on a partition on a regular basis. There are many tools available for this task; Management Central is probably the easiest one for monitoring partition performance. See 6.2, “Management Central monitors” on page 130 for detailed information about using this tool.

2.6 Collect the data

In addition to monitoring the performance, you need data in order to analyze partition performance. This data can be collected in two ways:

- Collecting data with Command Language (CL) commands
- Using Collection Services to collect data

There are commands available that enable you to both monitor the partition performance and collect the data. These commands and their use are discussed in 7.1.3, “Using CL commands for trend analysis” on page 185.

We recommend using Collection Services on a 24x7 basis. This enables you to choose the data that fits your needs instead of having only a small amount of data that may or may not be a good representation of the workload on the partition. Using Collection Services is discussed in detail in 4.2, “Collection Services” on page 48.

2.7 Analyze the data

This is the part of the Performance Management Cycle that requires the most skills and experience. You need to be able to find the essential data from potentially large amounts of information being collected. Look for the fluctuations (peaks and valleys in the graphs) in the resource utilizations and discover what the business cycle is.

Compare the latest reports with the baseline and decide if there is a performance problem. If the partition is suffering from a performance problem, discover whether the problem is a temporary one or has the performance degraded gradually as the workload has been increasing.

If the problem has been around for some time, when did it start? Was it after adding a new part of the application code, or after upgrading the application level? Have you upgraded the operating system or any other licensed program products?

A good thing to remember is that performance problems always have a root cause and do not occur for no reason.

2.8 Remove the identified performance problem

Is the hardware functioning properly? One example of a bad performance degradation is what happens when the cache on a disk controller becomes disabled. All the write operations slow down radically while the only visible symptoms are an entry in the QSYSOPR message queue and the disk units, which show a state of degraded on the WRKDSKSTS panel. The Work with Problem (WRKPRB) command is also useful for verifying the hardware’s status.

A performance problem caused by faulty hardware is easily corrected by having the hardware repaired or replaced.

If the problem is caused by software, the solution is usually more difficult to discover. First, you should verify that you are running the current levels of licensed program products by comparing the Work with PTF Groups (WRKPTFGRP) display to the information provided by the IBM services. The next step is to verify that the application programs are at the most current levels and update them if necessary.

Sometimes the slow growth of the workload goes unnoticed until there is a failure. To prevent this situation, you need to perform trend analysis periodically.

2.9 Tune the partition

The primary aim of performance tuning is to make the most efficient use of the partition’s resources. Performance tuning is a way to adjust the performance of the partition either manually or automatically. Each partition environment is unique in that it requires you to observe the partition’s performance and make adjustments that are best for your environment; in other words, you must do routine performance monitoring.

In many cases, tuning the performance of a partition can improve the overall performance level. A good starting point for performance tuning is described in the Information Center found at the following address:

http://publib.boulder.ibm.com/infocenter/iseries/v6r1m0/index.jsp

Select **Systems management** → **Performance** → **> Managing system performance**.

Refer to Chapter 8, “Performance Management best practices” on page 215 for guidelines and a quick start guide.

However, there comes a point in time when there are no more performance improvements to gain by tuning the partition. At that point, you need to periodically perform a trend analysis.

2.10 Analyze the trend

Trend analysis contains a review of performance data collected over a period of time and looks for relationships or ratios between throughput, response time, and the utilization of the various system resources. All these variables may impact the agreed level of performance.

By analyzing trends, you should be able to see the pattern of your system workload changes, know whether they are steady, increasing, or decreasing, and know how much of the resources reserved for the partition are being used.

You need to perform trend analysis periodically to make sure that you will not have unplanned outages caused by a gradually growing workload or by a completely filled disk subsystem. Whether the trend grows or sinks, the rate of change is the most important factor.

No matter which method you use for collecting data, you must make sure that you analyze data sets that are collected during the same phase of your business cycle. If you compare data collected during a time of low activity to data collected during the weekly peak of operations, you are not making a valid comparison. You should compare the time from a peak period to a peak period.

Trend analysis should be done separately for interactive and batch environments. If your company uses a certain application extensively, you may want to perform a trend analysis for the application.

Another item that may be important to track would be the end-of-month processing.

It is important that you collect trend analysis data consistently. If your system’s peak workload hours are between 10:00 a.m. and 2:00 p.m. and you collect trend analysis data for this time period, do not compare this data to data collected from other time periods.

To do a proper job of trend analysis, you must collect, analyze, maintain, and archive performance data. IBM Performance Management for Power Systems is a set of tools that can help you with your trend analyzing process.

2.11 Plan for capacity changes

Capacity planning is the process of determining the production capacity needed by an organization to meet the changing demands for its products. When discussing capacity planning for a partition, *capacity* refers to the maximum amount of work that a partition can manage in a given period of time.

The difference between the capacity of a partition and the required workload results in inefficiency, either in underutilized resources or unfulfilled user requests. The goal of capacity planning is to minimize this difference or remove it completely.

The IBM Systems Workload Estimator (WLE) is a web-based sizing tool for IBM Power Systems, IBM System i, IBM System p®, and IBM System x®. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems.

The WLE allows the measurement input to best reflect your current workload; one method is by using data from Performance Management for Power Systems. The WLE also provides several built-in workloads to reflect your emerging application requirements. The WLE provides current and growth recommendations for processor, memory, and disk that satisfy the overall client performance requirements.

Using the WLE is described in Chapter 7, “Trending and capacity planning tools” on page 169.

2.12 Which tools to use during the cycle

There are many IBM supplied tools that you can choose to use. Some of the tools are shipped with the operating system, and some of them you need to buy separately. Some of the tools can be used during multiple steps of the performance management life cycle while some of them are suitable for one specific step only.

2.12.1 IBM Systems Director Navigator for i performance interface

IBM Systems Director Navigator for i is the new Web-based console that was introduced with IBM i V6.1. All you need to use this tool is a browser that you point to your IBM i partition a the following address:

```
http://systemname:2001
```

Sign on with your user ID, enter your password, and start exploring.

> **Note:** You need to have both ports 2001 and 2005 enabled in your firewall rules to use the IBM Systems Director Navigator for i on your partition.

IBM Systems Director Navigator for i Performance interface (Performance Data Investigator) allows you to display and manage performance data by bringing together a variety of performance metrics into one unified whole. The Performance interface is a part of the IBM Systems Director Navigator for i. The Performance interface allows you to display summarized data in many different types of charts and tables. This can provide you with a quick picture of what is happening on your system. The interface allows you to start with a summarized view and then drill down into the data for more details.

Using the Performance interface is discussed in Chapter 5, “Performance data analysis” on page 81 and in the Information Center. Go to the IBM i V6.1 Information Center and **Systems management** → **Performance** → **Applications for performance management** in the Information Center to access information about the Performance interface.

Note: This book is based on the October, 2009 enhancements, and for this reason some PTFs are needed. They are SI35659, SI35663, and SI35833.

2.12.2 Collection Services

Collection Services is the primary tool to collect system performance data on IBM i.

Collection Services collects data that identifies the relative amount of system resources used by different areas of your system. Use Collection Services to:

- Easily manage your collection objects.
- Collect performance data continuously and automatically with minimal system impact.
- Control what data is collected and how the data is used.
- Move performance data between releases without converting the data.
- Create performance data files.
- Integrate your own programs to collect user-defined performance data into Collection Services.

Control Services can be configured with CL (Control Language commands) or using the Web interface.

2.12.3 IBM Performance Management for Power Systems

Behind any successful business, there is an information technology (IT) system at work., hopefully one that is operating at its peak efficiencies and benefiting from both effective capacity planning and performance analysis.

In today's business on demand era, performance and capacity management are more critical than ever. Systems that are available without disruption are essential. You must be able to understand what capacities are needed to capitalize on new business opportunities that possibly require the use of new software technologies or additional hardware.

The IBM Performance Management for Power Systems service offering automates the collection, archival, and analysis of system performance data and provides you with reports to help you manage system resources and capacity.

Based on your collected and summarized PM for Power Systems data, you can easily size your next system upgrade. For no additional charge and through the Web, you have an option to send your summarized PM for Power Systems historical growth data to the IBM Systems Workload Estimator (WLE) to get a preliminary recommendation about growth requirements for a system.

The PM for Power Systems reports can provide the information you need to:

- Forecast data processing growths based on trends.
- Plan and maximize return on hardware investments.
- Identify performance bottlenecks.
- Size the next system upgrade through integration with the WLE.

A Systems Management Guide to Performance Management for System i and System p servers, SG24-7122 provides detailed steps about using this service offering. The online version is available at the following URL:

http://www.redbooks.ibm.com/abstracts/sg247122.html?Open

2.12.4 System i Navigator Graph History function

A graph history provides a graphical display of performance data collected by Collection Services over a specified period of time. This interface is easy to use for both monitoring purposes and the trend analysis. See 7.1.2, “Using Navigator for i for trending” on page 177 for a detailed description about using this tool.

2.12.5 SQL performance Monitors

The System i Navigator version of the Start Database Monitor (STRDBMON) function is called a detailed SQL performance monitor. Start this monitor by right-clicking **SQL Performance Monitors** under the database portion of the System i Navigator tree and select **New** → **Detailed**. You can monitor a single query or all queries. After the monitor is started, it is shown in the SQL performance monitors list.

SQL Performance Monitors provide several predefined reports that you can use to analyze your data. To view these reports, right-click a monitor and select **Analyze**. The monitor does not have to be stopped in order to view this information. Additionally, you can import a monitor that has been started by using STRDBMON or some other interface. Right-click **SQL Performance Monitors** and select **Import**. You can then use the same predefined reports to analyze your monitor data.

SQL Visual Explain is a tool that is used to graphically display the optimization method selected by the database engine (the *optimizer*) when a query is executed. Visual Explain provides a graphical representation and makes it easier for you to determine actions to be taken (such as creating a new index).

Visual Explain can be used:

- On SQL statements stored in database performance monitor data
- On detailed monitor data (STRDBMON or Detailed SQL Performance Monitor)
- With the SQL Performance Monitors task in the System i Navigator
- On SQL statements in the SQL Script Center

Monitor data has to be collected at the OS/400 V4R5 or later level of the operating system to guarantee that all of the information is available for representing the data accurately. You may also collect data on one partition and analyze it on a different one. There is an excellent book about SQL performance, *OnDemand SQL Performance Analysis Simplified on DB2 for i5/OS in V5R4*, SG24-7326, that can be found at the following address:

http://www.redbooks.ibm.com/abstracts/sg247326.html?Open

2.12.6 Performance Tools for i

The Performance Tools for i product is a set of useful tools for viewing, analyzing, reporting and graphing performance data collected by Collection Services. This set of tools includes:

- Reports
- Commands
- Graphs
- Performance Tools plug-in for System i Navigator (also known as Display Performance Data graphical user interface)

Note: As of V6.1, the Performance Tools product can be used on Collection Services data and has GUIs for Disk Watcher and Job Watcher.

Usually, you collect performance data for trend analysis and performance analysis. The Performance Tools product offers you many tools to process, organize, and analyze your performance data. You can print a variety of reports that will organize performance data in a logical and useful format.

You can view performance data, display graphs to show trends, and analyze your system performance. For this purpose, you can use interactive commands like Display Performance Data (DSPPFRDTA), and Analyze Performance Data (ANZPFRDTA), or you can display performance data from the Display Performance Data graphical user interface (GUI), which is a plug-in for Navigator for i.

For more information about Performance tools for i, visit the Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Performance Tools**.

2.12.7 Job Watcher

Job Watcher provides for the collection of job data for any or all jobs, threads, and tasks on a partition. It provides information about call stacks, SQL statements, objects being waited on, Java JVM statistics, wait statistics, and so on, which is used to diagnose job related performance problems. The Job Watcher is similar in sampling function to the CL commands Work with Active Jobs (WRKACTJOB) and Work with System Activity (WRKSYSACT) in that, with each refresh, it computes delta information for the ending interval. The data collected from the jobs, threads, or tasks being watched is done in a non-intrusive manner.

The Job Watcher can be configured and managed through the IBM Systems Director Navigator for i Performance interface or CL commands.

2.12.8 Disk Watcher

Disk Watcher provides for the collection of disk performance data to diagnose disk related performance problems. The Disk Watcher allows you to obtain data concerning I/O operations to disk units, along with frequently needed runtime data to determine which objects, files, processes, threads, and tasks are being accessed. This tool brings up data beyond what is provided by such tools as Work with Disk Status (WRKDSKSTS), Work with System Status (WRKSYSSTS), and WRKSYSACT. Disk Watcher provides a mechanism to use short and longer duration traces to collect disk I/O data along with the associated task and object name.

Some potential uses of this tool are:

- Evaluating the performance of I/O operations on multi-path disk units
- Evaluating the performance of I/O queuing
- Determining how performance may be improved by re-spreading data across disk units
- Determining the optimal placement of devices, Input/Output Adapters (IOA), or buses

The Disk Watcher can be configured and managed through the IBM Systems Director Navigator for i Performance interface or CL commands.

2.12.9 Performance Explorer

Performance Explorer (PEX) collects extremely detailed information about a specific application, program, or system resource, and provides detailed insight into a specific performance problem. This includes the capability both to perform several types and levels of traces and to run detailed reports.

Performance Explorer is a data collection tool that helps the user identify the causes of performance problems that cannot be identified by collecting data using Collection Services or by doing general trend analysis. Two reasons to use Performance Explorer include:

- Isolating performance problems to the system resource, application, program, procedure, or method that is causing the problem
- Analyzing the performance of applications

Performance Explorer is a tool that helps find the causes of performance problems that cannot be identified by using tools that do general performance monitoring. As your computer environment grows both in size and in complexity, it is reasonable for your performance analysis to gain in complexity as well. The Performance Explorer addresses this growth in complexity by gathering data on complex performance problems.

Note: Only use PEX after you have tried the other tools. It gathers specific forms of data that can more easily isolate the factors involved in a performance problem; however, when you collect this data, you can significantly affect the performance of your system.

2.12.10 Using the appropriate tool

As mentioned earlier, there is a time and place for using each of these tools. An especially good tool can be in a variety of situations, as shown in Table 2-1.

Table 2-1 When to use which tool

Part of the management cycle	Tool(s) to use
Planning for the Performance Management Cycle	Recording instrument (word processor, pen and paper, and so on).
Setting the baseline	Meetings, Collection Services. WRKSYSSTS, WRKACTJOB. System i Navigator Monitors.

Part of the management cycle	Tool(s) to use
Collecting the data	Collection Services. Job Watcher, Disk Watcher. STRPFRCOL → CRTPFRDTA. WRKSYSSTS, WRKACTJOB, WRKSYSACT, WRKDSKSTS, and PEX.
Analyzing the data	Systems Director Navigator for i, iDoctor. WRKSYSSTS, WRKACTJOB, WRKSYSACT, and WRKDSKSTS. Performance tools for i reports: System Report and Transaction Report.
Deciding if there is a problem	Listen to the users. Systems Director Navigator for i and iDoctor. Performance tools for i reports: System Report and Transaction Report. WRKSYSACT and WRKDSKSTS.
Removing the problem	Fix the application. iDoctor. PEX.
Analyzing the trend	Performance Management for Power Systems. Comparing the self created Snapshots with each other. Analyzing self created graphs.
Planning for capacity	WLE.

Note: This table shows an example of using the tools. You may decide upon a different approach.

Part 2

IBM i tools

In this part, we discuss the IBM tools that are available for managing the performance of your IBM i partitions.

This part includes the following chapters:

- Chapter 3, “Overview of data collectors and tools” on page 29 introduces and positions the various data collection tools.
- Chapter 4, “IBM i Performance Data Collectors” on page 47 shows you how to use the collector tools.
- Chapter 5, “Performance data analysis” on page 81 shows you how to analyze the previously collected data.
- Chapter 6, “Real-time monitoring tools” on page 129 provides you with information about tools to use for monitoring the partition performance in real time.
- Chapter 7, “Trending and capacity planning tools” on page 169 shows you the tools for analyzing the performance trends and planning for capacity changes.

© Copyright IBM Corp. 2009. All rights reserved.

3

Overview of data collectors and tools

The key strengths of performance management on IBM i are the industry leading data collection technologies, which are built into the operating system, and the rich toolset that is available to analyze that data. The attributes of these industry leading capabilities include:

- Patented wait accounting technology.
- Integrated software that allows for seamless collection of metrics.
- Non-intrusive Collection Services that is on 24x7.
- Integrated database for performance data mining.
- Graphical interface for analyzing performance data.
- Performance that can be measured, monitored, and analyzed with various tools.

This chapter describes the data collection technology and then provides more information about the tools available with IBM i V6.1. It also provides guidance on which tool or combination of tools are best suited for a given task in the performance management life cycle.

© Copyright IBM Corp. 2009. All rights reserved.

3.1 Performance data collection overview

With IBM i V6.1, all of the data collection functions are integrated into the base operating system. IBM i provides all of the metrics needed by the IBM i data analysis tools.

The following sections discuss some of the key attributes of performance data collection on IBM i.

Note: Products that collect and analyze performance data from disparate operating systems may still require their own data collection functions. An example of this is IBM Tivoli Monitoring, which requires that an agent be installed on IBM i. This agent can then extract data from IBM i integrated collections or collect other data using IBM i application programming interfaces.

3.1.1 Built-in metrics

The integrated nature of the operating system allows IBM to build metrics into all *components* of IBM i and the Licensed Internal Code. In the context of this discussion, components of IBM i are functional areas of code, such as TCP/IP, Java Virtual Machine, IBM DB2, memory pools, LPARs, I/O processors, and so on. Applications can also generate their own metrics, as is the case with IBM products such as IBM Lotus® Domino and IBM WebSphere Application Server.

This is a unique feature because it allows IBM to produce standardized data across all facets of IBM i and minimize the need to use different analysis tools to analyze data from different components. It also allows Independent Software Vendors or customers to include metrics in their own applications, which can be included in the IBM i data collectors.

In IBM i V6.1, the data collectors include:

- Collection Services
- Disk Watcher
- Job Watcher
- Performance Explorer

Figure 3-1 shows these types of collectors.

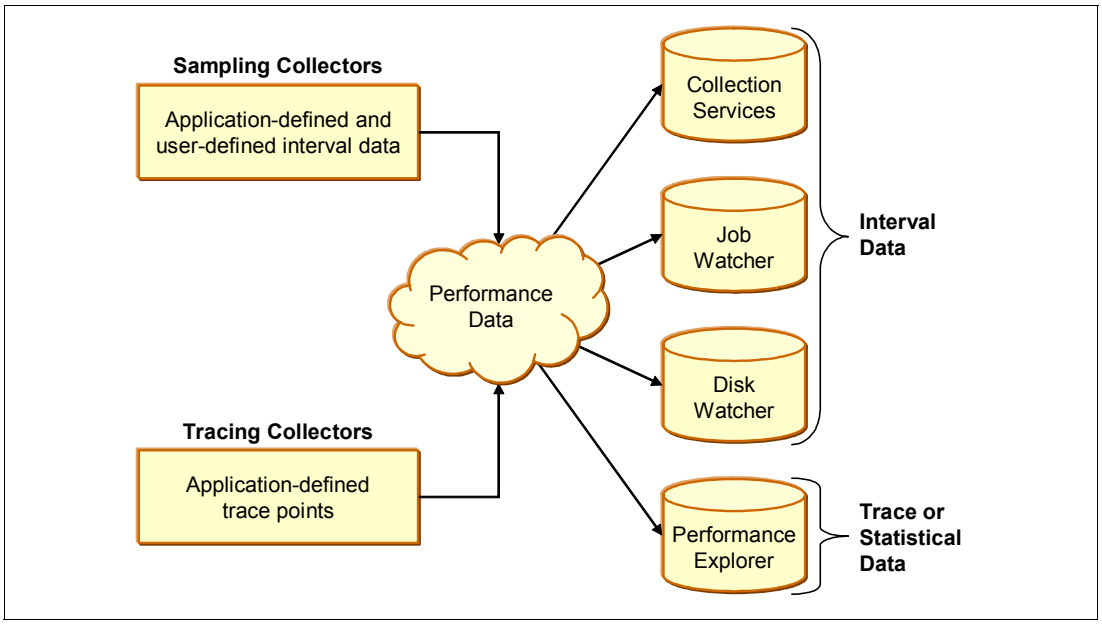


Figure 3-1 Collectors overview

An overview of each of these collectors, and their unique properties, are described in 3.2, “Performance data collectors overview” on page 34. For a more detailed discussion about the collectors, refer to 4.1, “Introduction to data collectors” on page 48.

In the majority of cases, the data collectors *harvest* the metrics from the operating system components. They do this on a *sampling* basis, where, at a nominated interval, they collect the metrics from the components and store it for subsequent processing by one of the data analysis tools.

In most cases, the collectors gather a standard set of metrics for all jobs and tasks on the partition. For some job types, additional data is also gathered. For example, the Collection Services collector gathers additional data for the follow job types:

- Communications jobs
- HTTP server jobs
- Java Virtual Machine jobs
- WebSphere Application Server
- Lotus Domino and others

3.1.2 Wait accounting

Another unique feature of IBM i performance data collection is the use of *wait accounting* technology. Wait accounting is the ability to determine what a job is doing when it is not running. In this context, a job is not running when it is active in the partition but is in a wait state. In general, there are two categories of waits, often described as *idle*, *good*, or *normal waits*, and *blocked*, *bad*, or *abnormal waits*.

Idle waits often occur when a job is awaiting input from a user, historically described as a key/think time. In this scenario, the job has executed some processing and is now in a wait state. The wait lasts for as long as the user is thinking about what data to enter, or what selection to take next, and for as long as it takes the user to enter such data or select an option. Similarly, idle waits can occur when an application is waiting for input from another partition.

Blocked waits can occur when a job has requested an operating system service but is either temporarily prevented from using the service or the service takes an excessive amount of time. Examples of blocked waits include:

- Waiting to access an object but the object is in use by another job that has an exclusive lock or seize on the object.
- Waiting for the database manager to return the requested data from a table or a file.

Blocked waits can add significantly to the processing time of a job. Table 3-1 provides a breakdown of the processing time of a job.

Table 3-1 Components of job processing time

State	Description
Processing in the CPU	The job has gathered all the information it needs and has entered the processor and is executing.
Queued for the CPU	The job has gathered all the information it needs and is now waiting in a queue for the processor.
Wait - Idle	The job is waiting for input from a user or another partition.
Wait - blocked	The job is requesting an operating system service but is temporarily prevented from using the service or the service is taking an excessive amount of time.

Wait accounting can help identify any bottlenecks causes by blocked waits. To illustrate this, we will use an example of a batch job that takes approximately 6 hours to process. The components of the job's processing time, known as its *run/wait signature*, is shown in Figure 3-2.

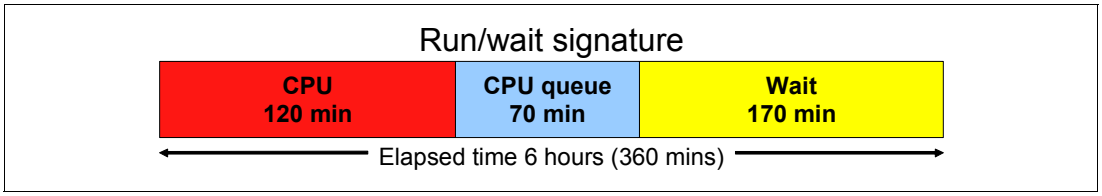


Figure 3-2 An example of a run/wait signature

Looking at this run/wait signature, we can see that the job could run in 3 hours and 10 minutes (CPU 120 minutes + CPU queuing 70 minutes), if we are able to eliminate the wait time.

Wait analysis allows us to delve deeper into the wait time and see what type of waits are occurring in our example batch job, as shown in Figure 3-3.

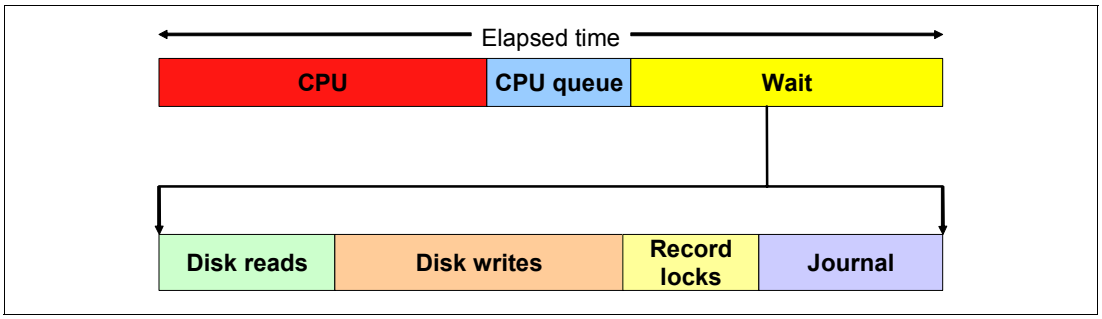
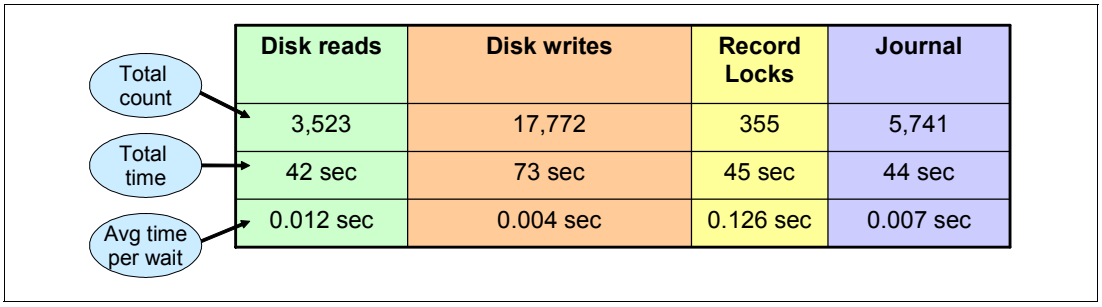


Figure 3-3 Breaking down wait time into components

In Figure 3-3, we can see that this job has different types of waits:

- Disk reads
- Disk writes
- Record locks
- Journal

In Figure 3-4, we show how, with wait accounting, we can drill down further and see the metrics of each wait component.



Disk reads	Disk writes	Record Locks	Journal
3,523	17,772	355	5,741
42 sec	73 sec	45 sec	44 sec
0.012 sec	0.004 sec	0.126 sec	0.007 sec

Figure 3-4 Detailing wait time: metrics

At this level, we can start to think about how we might go about reducing either the number of waits or the average length of each wait. For example:

- How many of the reads are page faults? Could memory/pool changes help?
- What programs are causing reads? Could they be reduced or made asynchronous?
- What programs are causing writes? Could they be reduced or made synchronous?
- What files are involved with the record locks? Can we eliminate the conflicts?
- What files are being journaled? Are journals needed and optimally configured?

Wait analysis and reduction can be a powerful and cost-effective way to improve response time and throughput.

3.1.3 Storing collected data

The data harvested by the collectors is ultimately stored in DB2 for i databases. The integrated databases allow the analysis tools to perform advanced performance data mining. The performance data can be saved and restored like any other database, either on the same partition or between partitions. For example, data collected on a production partition can be moved to another partition for analysis, minimizing any potential impacts to normal business operations on the production partition.

Users are also able to develop their own queries or reporting over these databases.

3.2 Performance data collectors overview

The following sections discuss, at a high level, some of the key differences between the data collectors. The collectors discussed are:

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer

Figure 3-5 shows the relationship between the collectors. Each collector differs in the level of detail collected, as well as the type of data collected.

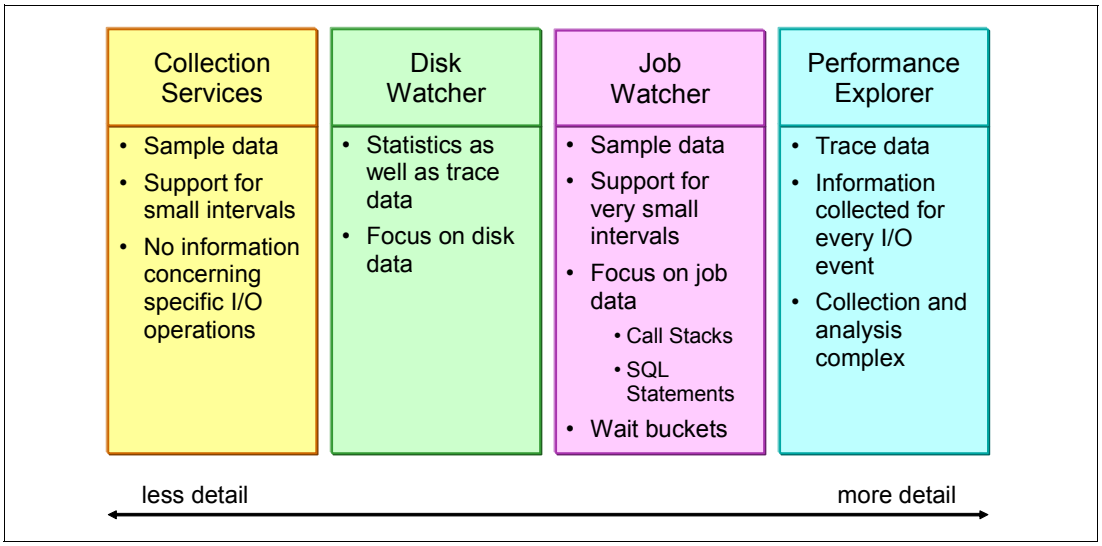


Figure 3-5 Positioning the performance data collectors

3.2.1 Collection Services collector

Collection Services provides for the collection of partition management data. It is the primary collector of system data.

Collection Services collects data that identifies the relative amount of partition resource used by different areas of a system. Collection Services can be used to:

- Manage your collection objects.
- Collect performance data continuously and automatically with minimal resource impact.
- Control what data is collected and how the data is used.
- Create performance data files.
- Integrate your own programs to collect user-defined performance data into Collection Services.

The Collection Services collector, by default, is always on. The default settings instruct Collection Services to automatically manage performance data collections on a daily basis, each day at around midnight, Collection Services will close out the collection for the day and start a new one for the next day. This process of ending one collection and starting a subsequent one is called “cycling the collection”. This is done to ensure that no single collection becomes too large to manage. The collection can also be ended manually at any point in time.

Collection Services uses a sampling methodology to collect performance data. The default sampling interval is set to 15 minutes. This means that every 15 minutes, the collector harvests the metrics from the IBM i components and stores them in a *management collection* object. A management collection (*MGTCOL) object is an efficient way of storing the raw performance data. The Create Performance Data (CRTPFRDTA) command can be used to convert the raw performance data into the DB2 for i database files. Figure 3-6 shows the relationship between the collector, the management collection object, and the database.

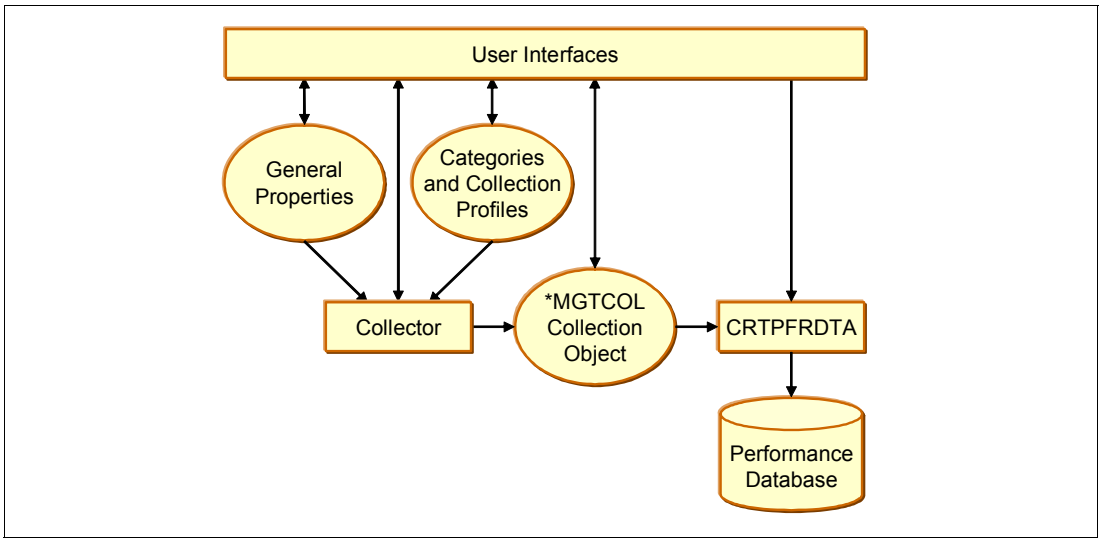


Figure 3-6 Collection Services collection overview

In most cases, collecting data at 15 minute intervals captures enough data to determine whether performance problems are occurring, with minimal impact on a partition. However, some of the collected metrics are averaged for a given interval, which can cause peaks or valleys of usage to not be noticed within the interval, so it may be necessary to collect data at shorter intervals. Shorter intervals produce a bigger impact, but due to the unique design, as discussed in 3.1.1, “Built-in metrics” on page 30, the impact is minimized.

For further details, refer to 4.2, “Collection Services” on page 48.

3.2.2 Job Watcher collector

Job Watcher provides for the collection of job data for any or all jobs, threads, and tasks on the partition. It provides call stacks, SQL statements, objects being waited on, Java Virtual Machine (JVM) statistics, wait statistics and more, which are used to diagnose job related performance problems.

Job Watcher is similar in sampling function to the CL commands WRKACTJOB and WRKSYSACT, in that each refresh computes delta information for the ending interval. The data collected from the jobs, threads, or tasks being watched is done so in a non-intrusive manner.

Job Watcher can be configured and managed through the IBM Systems Director Navigator for i Performance interface, CL commands, or IBM iDoctor for IBM i Job Watcher.

Job Watcher is not started automatically. It must be started explicitly, specifying the sampling interval and the duration of the collection. It can be ended automatically or manually. Job Watcher monitors can be set up to collect data continuously.

Job Watcher does not use an intermediary management collection object; it writes the collected data to a DB2 for i database as it collects the metrics.

For further details, refer to 4.3, “Job Watcher” on page 59.

3.2.3 Disk Watcher collector

Disk Watcher provides for the collection of disk performance data to diagnose disk related performance problems.

Disk Watcher allows you to obtain data concerning I/O operations to disk units, along with frequently needed runtime data to determine which objects, files, processes, threads, and tasks are being accessed. This tool surfaces data beyond what is provided by such tools as WRKDSKSTS, WRKSYSSTS, and WRKSYSACT. Disk Watcher provides a mechanism to use short and longer duration traces to collect disk I/O data along with the associated task and object name.

Some potential uses of this tool are:

- Evaluating the performance of I/O operations on multi-path disk units
- Evaluating the performance of I/O queuing
- Determining how performance may be improved by re-spreading data across units
- Determining the optimal placement of devices, IOAs, or buses

Disk Watcher can be configured and managed through the IBM Systems Director Navigator for i Performance interface, CL commands, or IBM iDoctor for IBM i Job Watcher - Disk Watcher.

Like Collection Services and Job Watcher, Disk Watcher uses sampling to collect statistical data. However, it also has a trace option that can be used to collect metrics for every I/O operation.

Disk Watcher is not started automatically. It must be started explicitly, specifying the sampling interval and the duration of the collection. It can be ended automatically or manually.

Disk Watcher does not use an intermediary management collection object; it writes the collected data to a DB2 for i database as it collects the metrics.

For further details, refer to 4.4, “Disk Watcher” on page 62.

3.2.4 Performance Explorer collector

Performance Explorer is a data collection tool that helps the user identify the causes of performance problems that cannot be identified by collecting data using one of the other data collectors or by doing general trend analysis. It collects more detailed information about a specific application, program, or partition resource, and provides detailed insight into a specific performance problem. Two reasons to use performance Explorer include:

- Isolating performance problems to the partition resource, application, program, procedure, or method that is causing the problem
- Analyzing the performance of applications

Note: Performance Explorer is the tool you need to use after you have tried the other tools. It gathers specific forms of data that can more easily isolate the factors involved in a performance problem; however, when you collect this data, you can significantly affect the performance of your partition. Performance Explorer can collect data by sampling or by a detailed trace collection. How Performance Explorer collects the data is specified on the PEX definition.

This tool is designed for application developers who are interested in understanding or improving the performance of their programs. It is also useful for users knowledgeable in performance management to help identify and isolate complex performance problems.

Like Job Watcher and Disk Watcher, Performance Explorer must be explicitly started. It can also be ended automatically or manually; however, the automatic end does not support a duration timer.

Another key difference between Performance Explorer and the other collectors is the method it uses to collect and store the metrics. While the collection is active, it writes the collected data to a management collection object. When the collection is stopped, by default it will write the data to a DB2 for i database and delete the management collection object.

For further details, refer to 4.5, “Performance Explorer” on page 66.

The IBM Redbooks publication, *Application and Program Performance Analysis Using PEX Statistics on IBM i5/OS*, SG24-7457, provides examples of Performance Explorer functions and examples of Performance Explorer trace support. It can be found at the following address:

http://www.redbooks.ibm.com/abstracts/sg247457.html

3.3 Data analysis and monitoring tools

In this section, we review many of the IBM data analysis and monitoring tools that are available and provide guidance on when they should be used.

3.3.1 Relationship between collectors and tools

As previously mentioned, there are a number of different collectors that can be run. The data from each type of collector is sometimes used by more than one tool, as shown in Figure 3-7.

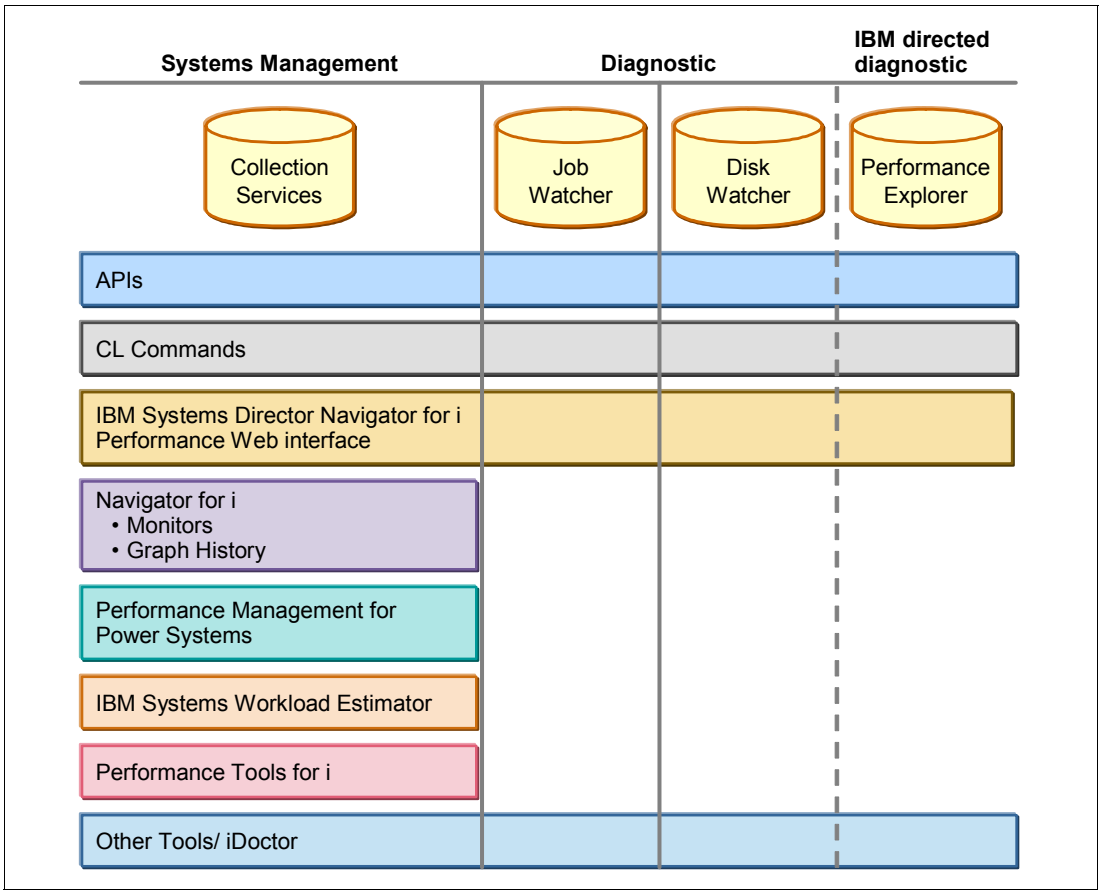


Figure 3-7 Performance data analysis tools and collected data

Figure 3-7 does not show the full list of tools; it is only for illustrative purposes. A more complete list of tools is provided in Table 3-2.

Table 3-2 Relationship between collectors and analysis tools

Collector	Analysis tool
Collection Services	IBM Systems Director Navigator for i: ► Health Indicators ► Collection Services
	Navigator for i: ► Monitors ► Graph History ► Performance Tools for i plug-in
	Performance Management for Power Systems
	IBM Systems Workload Estimator
	Performance Tools for i
	Collection Services Investigator

Collector	Analysis tool
Job Watcher	IBM Systems Director Navigator for i: Performance Data Investigator
	IBM iDoctor for IBM i Job Watcher
Disk Watcher	IBM Systems Director Navigator for i: Disk Watcher
	IBM iDoctor for IBM i Job Watcher: Disk Watcher
Performance Explorer	IBM Systems Director Navigator for i: Performance Explorer
	PEX Analyzer
	IBM i Print PEX Report (PRTPEXRPT) command

In general, most non-IBM performance analysis tools also use the data collected by Collection Services.

3.3.2 Tool usage

The tools available, particularly for analyzing Collection Services, can appear overwhelming. In addition, many tools can be used for more than one purpose. Table 3-3 describes the *intended* use of the different tools.

Table 3-3 Intended tool usage (IBM analysis tools)

Tool	Usage
IBM Systems Director Navigator for i: Performance Data Investigator (Health Indicators)	Used to provide a high level overview of key resource utilization metrics based on predefined thresholds. IBM i ships with default thresholds, but they can be customized. It is intended as a simple way to identify if one or more resources are overcommitted during a given collector period, which may indicate that performance may not be meeting user expectations.
IBM Systems Director Navigator for i: Collection Services	Used to present performance data for trend analysis and performance analysis. It has graphical tools to process, organize, and analyze performance data. It can be used to view performance data, display graphs to show trends, and analyze a partition's performance. It is also used to identify job related performance problems. The drilldown features can be used to help diagnose a given job related performance problem.
IBM Systems Director Navigator for i: Job Watcher	Used to identify job related performance problems. The drilldown features can then be used to help diagnose a given job related performance problem. It provides additional information not available in other tools, such as call stacks, SQL statements, objects being waited on, Java JVM statistics, wait statistics, and so on.

Tool	Usage
IBM Systems Director Navigator for i: Disk Watcher	Used to diagnose disk related performance problems. It can be used to identify any peaks in resource utilization of disk pools, disk units, and disk paths. It can also be used to identify which object is being used on certain disks and by which applications.
IBM Systems Director Navigator for i: Performance Explorer	Used to trace the flow of work in an application and the resource usage within application programs. It can then be used to drill down into the resource utilization of specific programs. It is intended to be used as directed by IBM.
Navigator for i: Monitors	Monitors are used for real time monitoring. They are used to track the elements of partition or job performance. The system monitors can graphically display in real time the changes to a particular performance metric over time, up to approximately the last 60 minutes. The Job monitors can monitor some job performance metrics as well as certain job status changes, such a job starting, ending, or being held.
Navigator for i: Graph History	Used to observe the variability of specific performance metrics over several months. It can be used to graphically view summarized collection services data over a 7 day period.
Navigator for i: Performance Tools for i plug-in	Used to present performance data for trend analysis and performance analysis. The Performance Tools for i plug-in offers graphical tools to process, organize, and analyze performance data. It can be used to view performance data, display graphs to show trends, and analyze a partition’s performance.
PM for Power Systems	Used to provide information about a partition's current and long term utilization trends and helps provide insight on where utilization is headed, what additional capability a partition has, and what upgrades might be needed for additional workloads. Historical performance data is made available so that utilization and capacity environment can be reviewed from up to 24 months prior.
Workload Estimator	Used to size a new partition, to size an upgrade to an existing partition, or to size a consolidation of several partitions. The Workload Estimator allows measurement input to best reflect current workloads and provides a variety of built-in workloads to reflect emerging application requirements. It supports sizings dealing with multiple systems, multiple partitions, multiple operating systems, and multiple time intervals. The Estimator also provides the ability to do multiple sizings.

Tool	Usage
Performance Tools for i	Used to present performance data for trend analysis and performance analysis. It can print a variety of reports that will organize performance data in a logical and useful format. It can be used to view performance data, display graphs on a 5250 display station to show trends, and analyze a partition's performance.
IBM iDoctor for IBM i: Collection Services Investigator	Used for the same purposes as the corresponding component in IBM Systems Director Navigator for i. However, it has more features and is more complex to use and is intended for use by experienced performance analysts.
IBM iDoctor for IBM i Job Watcher	Used for the same purposes as the corresponding component in IBM Systems Director Navigator for i. However, it has more features and is more complex to use and is intended for use by experienced performance analysts.
IBM iDoctor for IBM i - Disk Watcher	Used for the same purposes as the corresponding component in IBM Systems Director Navigator for i. However, it has more features and is more complex to use and is intended for use by experienced performance analysts.
IBM iDoctor for IBM i Performance Explorer	Used for the same purposes as the corresponding component in IBM Systems Director Navigator for i. However, it has more features and is more complex to use and is intended for use by experienced performance analysts.

In addition to the IBM i specific tools, IBM provides multi platform monitoring and analysis tools. These tools are intended for organizations that wish to manage multiple operating systems from a single point. These tools are not discussed in detail in this book. For more information about these products, refer to the following Web pages:

- Tivoli software:

  http://www.ibm.com/software/tivoli/

- IBM Systems Director:

  http://www.ibm.com/systems/management/director/

> **Note:** In this publication, we discuss IBM Systems Director Navigator for i in detail. This is not the same product as IBM Systems Director. IBM Systems Director Navigator for i is specific to IBM i, while IBM Systems Director is for managing multiple operating systems.
>
> Both products have similar user interfaces and there is a plug-in for IBM Systems Director Navigator for i into IBM Systems Director, but they are separate products.

Table 3-4 provides a summary of the key attributes of each of the tools in the IBM product set.

Note: In Table 3-4, Historical data refers to data that is retained for months or even years. Real time data refers to accessing current data and also, for collectors that are samplers and are active, data collected at the last interval.

Table 3-4 Summary of tool features

Tool	Sample data	Trace data	System level data	Job level data	Object level data	Historical data	Real time data	Web based interface	Drilldown
Performance Data Investigator Health Indicators	✓	X	✓				✓	✓	[a]
IBM Systems Director Navigator for i: Collection Services	✓	X	✓	✓		✓	✓	✓	✓
IBM Systems Director Navigator for i: Job Watcher	✓		✓	✓	✓		✓	✓	✓
IBM Systems Director Navigator for i: Disk Watcher	✓		✓	✓	✓		✓	✓	✓
IBM Systems Director Navigator for i: Performance Explorer	✓	✓	✓	✓	✓		N/A	✓	✓
Navigator for i: Monitors	✓		✓	✓[b]			✓		
Navigator for i: Graph History	✓		✓	✓		✓	✓		
Navigator for i: Performance Tools for i plug-in	✓		✓	✓			✓		✓
PM for Power Systems	✓		✓	✓		✓		V	X[c]
Workload Estimator	N/A	N/A	✓			✓		✓	
Performance Tools for i	✓		✓			✓	✓		✓
IBM iDoctor for IBM i Job Watcher - Collection Services Investigator	✓		✓	✓			✓		✓
IBM iDoctor for IBM i Job Watcher	✓		✓	✓	✓		✓		✓
IBM iDoctor for IBM i Job Watcher - Disk Watcher	✓		✓	✓	✓		✓		✓
IBM iDoctor for IBM i Performance Explorer		✓	✓	✓	✓		N/A		✓

a. Health Indicators has some limited drilldown capabilities.
b. Basic job information is available with System monitors. More detailed job information is available in Job monitors.
c. PM for Power Systems has some limited drilldown capabilities.

To assist readers of this publication with selecting an appropriate tool for a given performance situation, Table 3-5 provides some example situations. For each of these situations, a suggestion is made about which tools may be appropriate. However, it is only a guide. Other tools can also be used in a given situation and can produce similar diagnostic results. The choice of tool can also be dependent on how much background information can be gathered for a particular situation, and how proficient the user is with each of the tools.

In many cases, it will be a top-down approach. Tools that assist analyzing partition wide data are used initially to determine what area may require further investigation. The nature of that area usually dictates which other tool or tools may be needed to fully diagnose the cause.

Table 3-5 Selecting an appropriate tool

Symptom	Tool(s)
Receiving reports of poor response time from many users. This has been an ongoing problem for days or even weeks. Poor response times have been reported to be consistent throughout the day.	1. PM for Power Systems: This would be a useful tool to determine the utilization of key resources over time, and to try and determine whether any resources became overcommitted around the time the response time started to slow. 2. IBM Systems Director Navigator for i Collection Services: If any of the partition resources appeared overcommitted or heavily utilized in step 1, use this tool to try and determine why. 3. IBM Systems Director Navigator for i Job Watcher: If any of the partition resources did not appear to be overcommitted or heavily utilized in step 1, use this tool to try and determine if waits are part of the problem. If particular jobs appear to be using excessive resources, use this tool to try and identify why. 4. IBM Systems Director Navigator for i Disk Watcher: If a particular disk or disks appear to be used excessively or appear slow in step 2, use this tool to try and identify why.
An application developer has written a new program and, when testing, discovers it is taking longer than expected to run.	1. IBM Systems Director Navigator for i Performance Explorer: In this scenario, it is already clear which program is likely to be at fault, so Performance Explorer may be a suitable starting point. 2. iDoctor for PEX.
Users report that, for brief periods during the day, response times are unacceptable.	IBM Systems Director Navigator for i Collection Services: Use this tool to determine whether the times the problems are reported coincide with any unusual resource utilization. If so, use this tool to drill down to diagnose what the cause may be.

Symptom	Tool(s)
New workloads are planned for an existing partition.	1. PM for Power Systems: This would be a useful tool to determine the current utilization of key resources and also to identify trends. 2. PM for Power Systems combined with Workload Estimator: If the new workloads are coming from an exiting partition, PM for Power Systems can be used to import both workloads into Workload Estimator. 3. PM for Power Systems combined with Workload Estimator: If the new workloads are not pre-existing, PM for Power Systems can be used to import the current partition workload into Workload Estimator and then add other workloads to model proposed workloads.
A new company has been acquired and the transaction loads in the existing applications will increase significantly.	1. PM for Power Systems: This would be a useful tool to determine the current utilization of key resources and also to identify trends. 2. PM for Power Systems combined with Workload Estimator: PM for Power Systems can be used to import both workloads into Workload Estimator. 3. Using PDI to send data to WLE by way of “Size Next Upgrade” is another option, if the application could be characterized by one day’s worth of activity.
Overnight batch processing is taking an unacceptable amount of time.	1. PM for Power Systems: This would be a useful tool to determine the utilization of key resources over time, and to try and determine whether any resources became overcommitted around the time the batch processing started to slow. 2. IBM Systems Director Navigator for i Collection Services: If any of the partition resources appeared overcommitted or heavily utilized in step 1, use this tool to try and determine why. 3. IBM Systems Director Navigator for i Job Watcher: If any of the partition resources did not appear to be overcommitted or heavily utilized in step 1, use this tool to try and determine if waits are part of the problem. If particular jobs appear to be using excessive resources, use this tool to try and identify why. 4. IBM Systems Director Navigator for i Disk Watcher: If particular disk or disks appear to be used excessively or appear slow in step 2, use this tool to try and identify why. 5. IBM Systems Director Navigator for i Performance Explorer: If one or more programs in the batch processing appear to be consuming excessive CPU in step 3, this tool may assist determine what parts of the program are consuming those resources.

Symptom	Tool(s)
No performance problems are being reported, but a major partition change, such as an operating system upgrade or application upgrade/release, is planned.	Ensure Collection Services collectors are currently running and that enough data is being retained to establish a pre-upgrade baseline. Consideration should also be given to establishing baseline collections for the other collectors. If performance problems are reported after the upgrade, the baseline collections can be compared to the current collections using the appropriate tool to identify where the variances are.
Users in one location report consistently slow response times while users in other locations do not report these slow response times.	1. IBM Systems Director Navigator for i Collection Services: If the location is on a different network interface than the other locations, use this tool to identify any potential problems in traffic volumes. 2. Navigator for i Performance Tools for i plug-in: If the traffic volume appears high in step 1, the Navigator for i Performance Tools for i plug-in can be used to identify excessive utilization in communication resources.
A company is preparing a 3 year financial forecast and wants to get an indication of likely IT spending on additional capacity over that period.	1. PM for Power Systems: This would be a useful tool to determine the current utilization of key resources and also to identify trends. 2. PM for Power Systems combined with Workload Estimator: PM for Power Systems can be used to import both workloads into Workload Estimator. Growth estimates can be added to Workload Estimator to estimate future upgrade requirements.

3.3.3 Tool packaging

As previously discussed, the performance data collectors are included in the base operating system. However, the analysis and monitoring tools are typically packaged with a licensed program product. Table 3-6 provides the product name for each of the tools discussed in this chapter.

Table 3-6 Tool packaging

Tool	Product / option
IBM Systems Director Navigator for i: Health Indicators	Base operating system.
IBM Systems Director Navigator for i: Collection Services	Base operating system.
IBM Systems Director Navigator for i: Job Watcher	Performance Tools for i (5761PT1) - Option 3 - Job Watcher.
IBM Systems Director Navigator for i: Disk Watcher	Performance Tools for i (5761PT1) - Option 1 - Manager feature.
IBM Systems Director Navigator for i: Performance Explorer	Performance Tools for i (5761PT1) - Option 1 - Manager feature.
Navigator for i: Monitors	System i Access for Windows® (5761XE1).

Tool	Product / option
Navigator for i: Graph History	System i Access for Windows (5761XE1).
Navigator for i: Performance Tools for i plug-in	System i Access for Windows (5761XE1) and Performance Tools for i (5761PT1) - Option 1 - Manager feature.
PM for Power Systems	Base operating system (5761SS1).
Workload Estimator	Not applicable. Workload Estimator does not run on IBM i. It is a Web-based tool that can accept summarized Collection Services data from IBM i.
Performance Tools for i	Performance Tools for i (5761PT1) - Option 1 - Manager feature.
IBM iDoctor for IBM i: Collection Services Investigator	IBM iDoctor for IBM i.
IBM iDoctor for IBM i Job Watcher	IBM iDoctor for IBM i.
IBM iDoctor for IBM i: Disk Watcher	IBM iDoctor for IBM i.
IBM iDoctor for IBM i Performance Explorer	IBM iDoctor for IBM i - PEX Analyzer.

4

IBM i Performance Data Collectors

In this chapter, we discuss the Performance Data Collectors available on IBM i V6.1 and introduce you to the various interfaces that are available to manage the Performance Data Collections.

This chapter discusses the following collectors:

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer

We also discuss how to manage collections using the Performance interface in IBM Systems Director Navigator for i.

© Copyright IBM Corp. 2009. All rights reserved.

4.1 Introduction to data collectors

The base operating system on IBM i includes all the instrumentation needed for collecting performance data. There are four data collectors that have unique characteristics and offer specific insight into system performance:

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer

4.2 Collection Services

Collection Services is an IBM i function that is used for collecting System Management data. It can be used to gather performance data continuously and automatically with minimal system impact. The Collection Services interval data can be application defined or user defined and can be managed and analyzed using multiple interfaces.

4.2.1 How Collection Services works

Collection Services collects system and job level data at regular intervals that can be set from 15 seconds to 1 hour. At every sampling interval, Collection Services probes and gathers metrics from various system resources, including jobs, disk units, IOPs, buses, pools, and communication lines.

The raw performance data is stored in a single management collection object that takes advantage of teraspace support to make it a more efficient way to store large quantities of performance data. This also supports a release independent design that allows the moving the performance data to a system at a different release level without requiring database file conversions.

To use the performance data for analysis, performance database files must be generated using the CRTPFRDTA CL command. Starting with V6.1, database performance files are created by default in a low priority CRTPFRDTA job as the data is being collected to facilitate real-time analysis. In releases prior to V6.1, you could have the performance data files created automatically by configuring Collection Services to create the database files during the collection.

Independent management of data collection and performance database file generation facilitate the most efficient way of collecting data from large number of jobs and tasks at frequent intervals. The associated impact is minimal enough to justify running collection services continuously in the background and capture data needed to analyze and solve performance issues before they turn into a serious problem.

Collection Services does not provide Object and Program level details such as Call Stacks, Activation Groups, Locks, and SQL statements. Despite these limitations, Collection Services is useful for an overview of how a system has been performing over longer periods of time.

Performance Data Investigator within the web-based IBM Systems Director Navigator for i provides exclusive views with many charts and drill downs that are available with Collection Services data. Collection Services Investigator within IBM iDoctor for IBM i GUI also offers similar views of wait statistics.

4.2.2 Configuring Collection Services

Collection Services comes with default configured options, but they can be changed to suit your environment and needs. Collection Services Configuration allows you to select the library where the data is stored, cycle time and frequency, collection profile, data retention, and so on. The configuration can be done using multiple interfaces:

- IBM Navigator for i (Windows client)
- IBM Systems Director Navigator for i (Web-based console)
- Configure Performance Collection (CFGPFRCOL) CL command
- Change Collection Services Attributes (QypsChgColSrvAttributes) API

Configuration options

Collection Services Configuration options can be divided into three main categories and options in each category are explained in the following sections.

Basic Collection Services attributes

Here is a list of general Collection Services Configuration options:

Default Collection Interval Default Collection sample interval.

Collection Retention Period Specifies how long the management collection objects are retained.

Collection Library The library where the management collection objects are stored. Performance data is stored in QPFRDATA library by default. The data retention policy for Management Collection objects and performance database files applies only to the current configured library.

Cycle Time Specifies the time when the current collection is ended and data is collected in a new management collection object.

Cycle Interval The elapsed time between cycles after which the current collection is ended and data is collected in a new management collection object by the Collection Services server job (QYPSPFRCOL).

Create database files Specifies whether or not to submit the low-priority companion job (CRTPFRDATA) to create performance database files and process the data in the current management collection object as it is collected.

Default Collection Profile Specifies the collector definition that determines what categories of data to collect and how often to collect them.

Performance Database file support attributes

Here are additional Collection Services attributes that support performance database files:

Standard Data Retention Specifies how long the standard data in performance database files within the configured library is retained.

Summary Data Creation Specifies whether or not to create additional performance database summary data to facilitate quicker analysis.

Management Central Monitor support attributes

Here are the different options that Management Central Monitor supports:

Graph Data Creation	Gives you the option to generate detailed graph data automatically into a *MGTCOL object with *PFRDTL attribute when the collection is cycled.
Graph Data Retention	Specifies how long the Management central graph data is retained on the system.
Historical Data Creation	Gives you the option to generate historic graph data automatically into a *MGTCOL object with *PFRHST attribute when the collection is cycled.
Historical Data Retention	Specifies how long the Management central historical graph data is retained on the system.

This configuration parameters are only available on System i Navigator.

Note: The Cycle and Data Retention attribute changes take effect immediately while others take effect when the collection is cycled. Refer to the help text for the CFGPFRCOL command for more information.

Configure using IBM Systems Director Navigator for i

Collection Services can be configured using the Performance interface in IBM Systems Director Navigator for i by performing the following steps:

1. Go to the IBM Systems Director Navigator for i page by pointing your browser to the IBM i's IP address or DNS name and specify the default port 2001 as follows:

   `http://<system>:2001/`

2. Enter your IBM i user ID and password to log in to IBM Systems Director Navigator for i.

3. In the left pane, expand the i5/OS management tree and click **Performance**, as shown in Figure 4-1. When the Performance tab is displayed for the first time, it will only show the Investigate Data and Collections options.

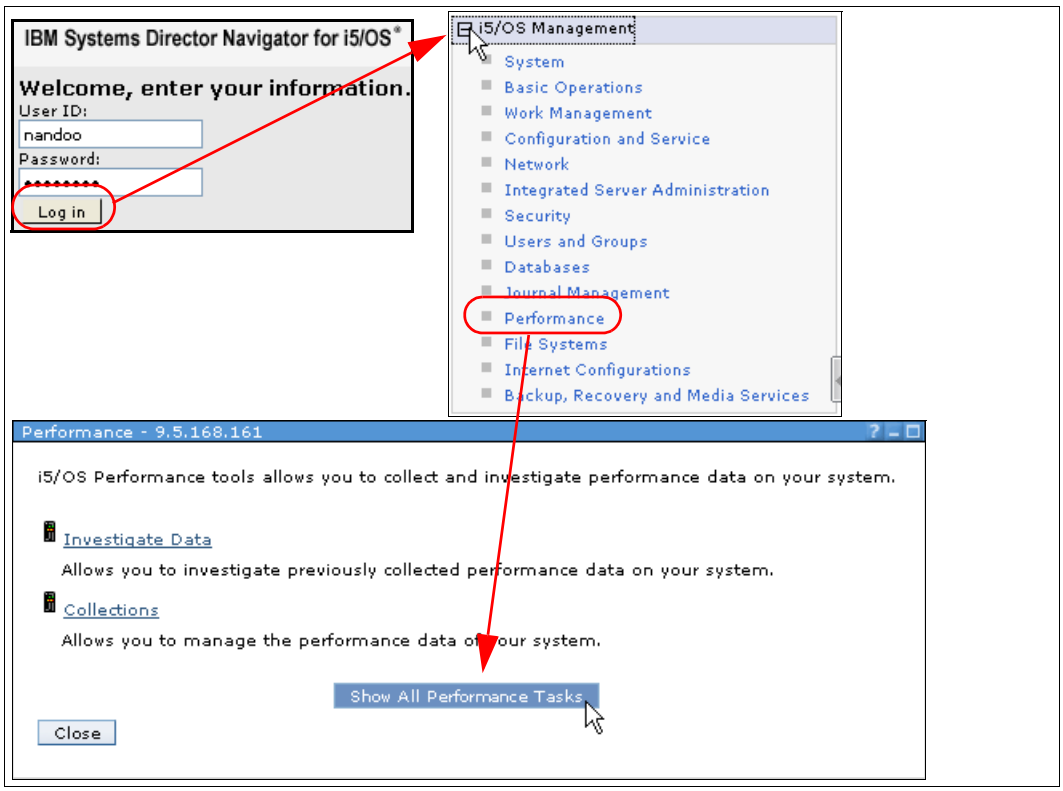


Figure 4-1 Navigate to the Show All Performance Tasks button

4. In the right pane, click **Show All Performance Tasks** to display all available Performance Management tasks.
5. As shown in Figure 4-2, expand **Collectors** and select **Collection Services** → **Configure Collection Services**

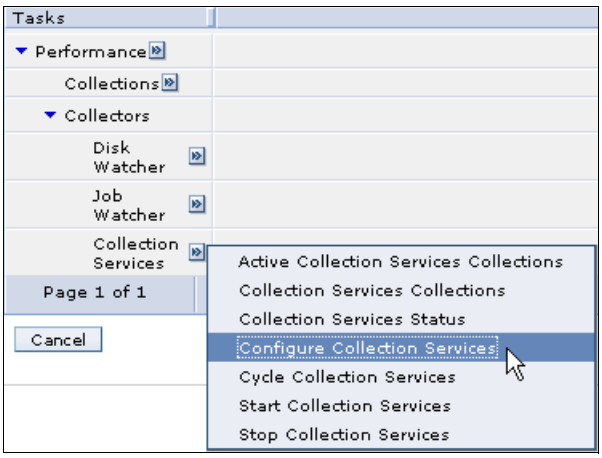


Figure 4-2 Configure Collection Services option

6. Make the appropriate changes in the three tabs, as shown on the Configure Collection Services window in Figure 4-3, and click **OK**.

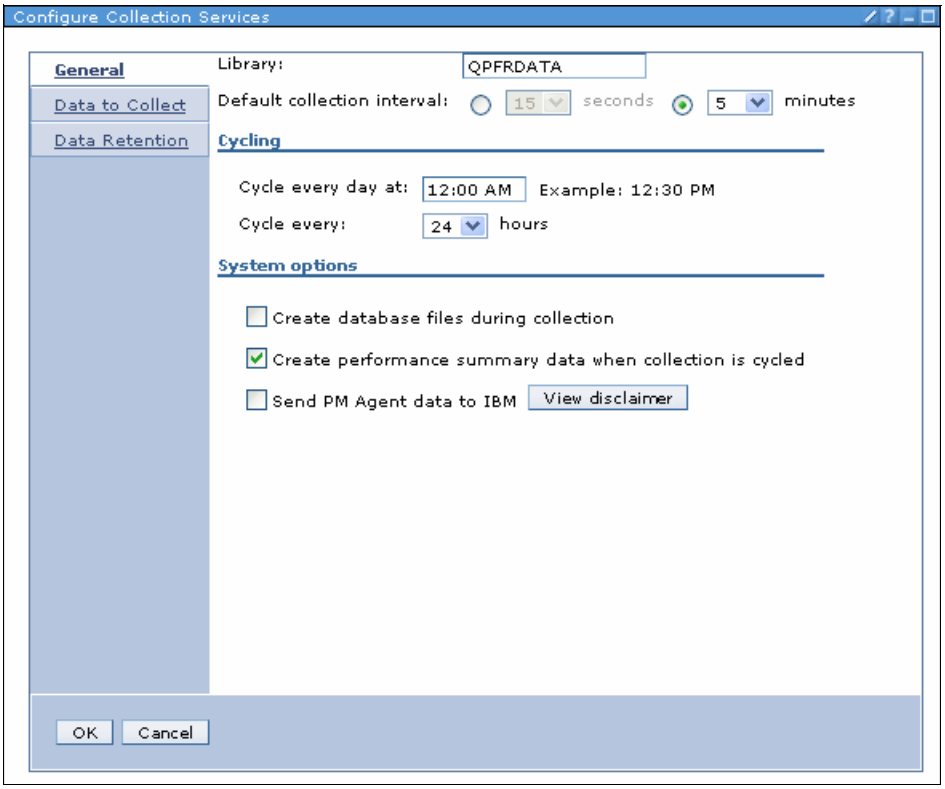


Figure 4-3 Configure Collection Services window

Configure using IBM Navigator for i

Collection Services can be configured using IBM Navigator for i by performing these steps:

1. In IBM Navigator for i, expand **Configuration and Service** under the system name and right-click **Collection Services**, as shown in Figure 4-4.

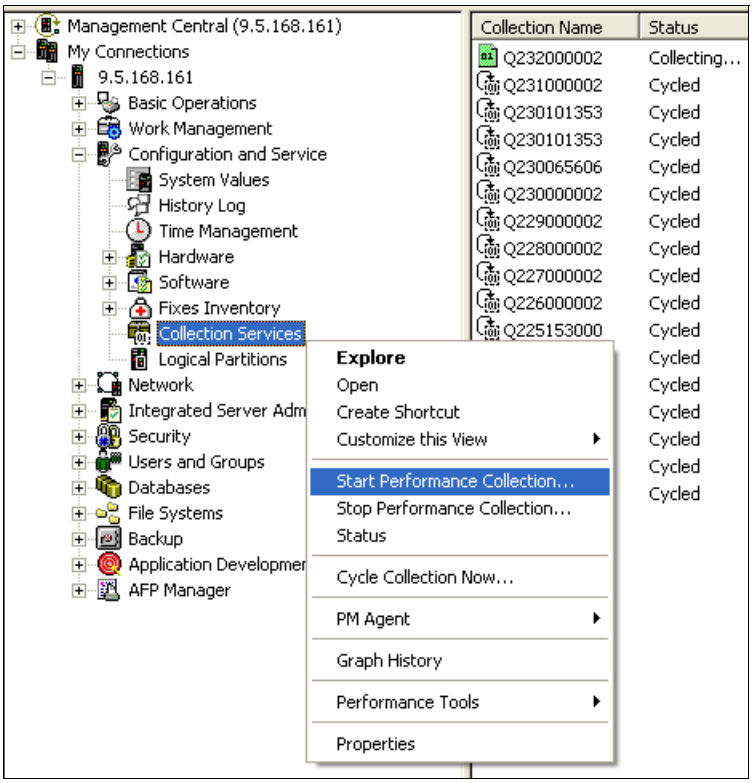


Figure 4-4 Collection Services context menu

2. In the Start Collection Services window, change the attributes as appropriate on the General tab and select the collection profile to be used on the Data to Collect tab, as shown in Figure 4-5.

> **Note:** Graph Data options are only available when using the IBM Navigator for i interface or the Change Collection Services Attributes API. Also, Graph data and Graph historic data creation options are only available for Central and End-point systems running at least OS/400 V5R1.

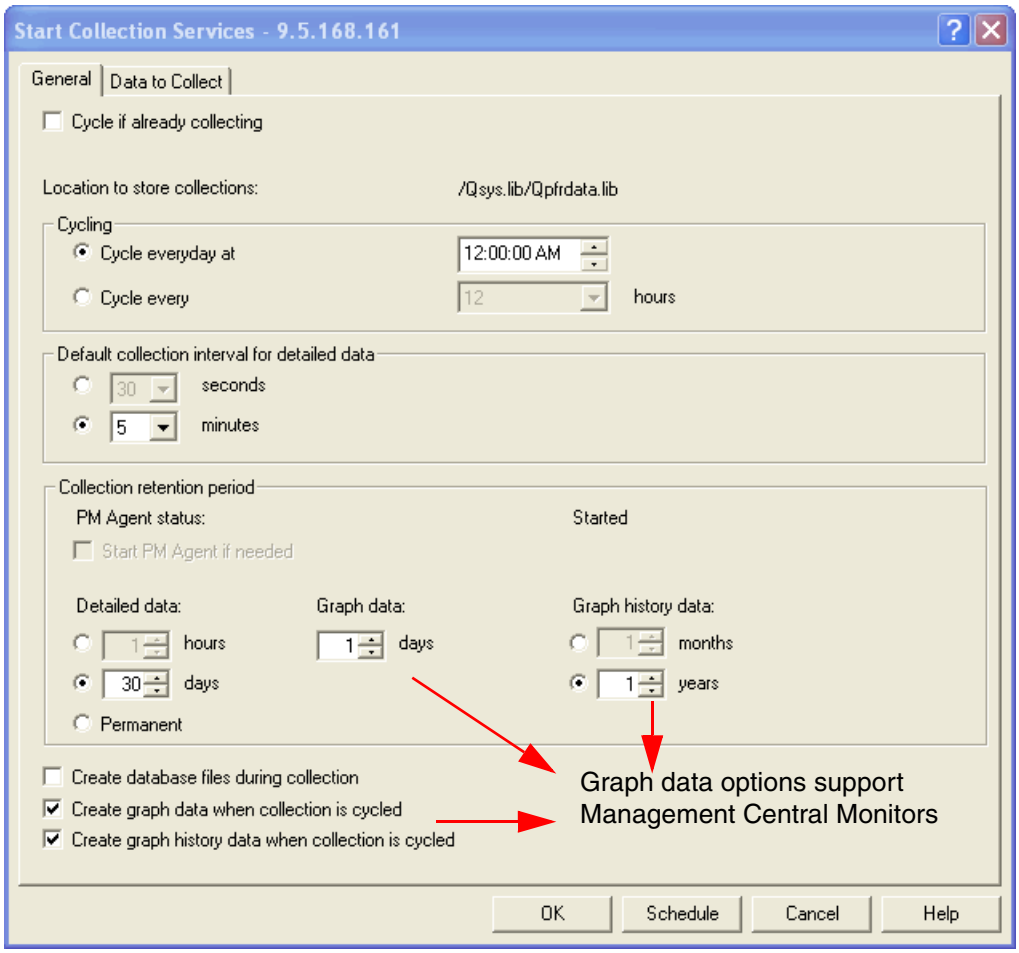


Figure 4-5 Start Collection Services window

4.2.3 Customizing Collection Services

The Data Collector Profile determines what data is collected and when it is collected. Collection Services provides four predefined Collection Profiles. However, Collection Services allows for controlling which categories of data are collected and how often they are collected by using a Custom Collection Profile.

Predefined Collection Profiles

Collection Services has four predefined IBM-supplied collection profiles that can be used to collect performance data from various data categories. Collection Services uses the Standard Plus Protocol profile by default to collect the performance data.

The four predefined IBM-supplied collection profiles are:

Minimum	This option collects the most commonly used performance data. The collection includes categories such as System Bus, Storage Pool, Hardware Configuration, System CPU, System-Level Data, Job MI, Job OS, Disk Storage, and IOP.
Standard	The data categories typically needed by the tools in Performance Tools for IBM i, except for communications protocol data. The data categories in the Standard profile include all the categories in the Minimum profile plus Storage Pool Tuning, Subsystem, SNADS Transaction, Local Response Time, APPN, SNA, TCP/IP Base, and User Transactions. The Domino category is included in this profile when the Domino for iSeries licensed program is installed on the system. The IBM HTTP Server (powered by Apache) category is included in this profile when the IBM HTTP Server for iSeries licensed program is installed on the system.
Standard Plus Protocol	The data categories typically needed by the tools in Performance Tools for IBM i, including communications protocol data. The data categories in the Standard plus protocol profile include all the categories in the Standard profile with the addition of Network Server, Communications Base, Communications Station, Communications SAP, Data port services, Logical partition, and TCP/IP Interface.
Enhanced Capacity Planning	The data categories in the Enhanced capacity planning profile include all the categories in the Standard plus protocol profile with the addition of PEX Data - Processor Efficiency. This option is not recommended.

For more information about Collection Profile Data categories, go to the IBM i V6.1 Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Collection Services** → **Customizing data collections** → **Collection Services collection profiles**.

Custom Collection Profile

Using the IBM Systems Director Navigator for i Web Console or IBM Navigator for i Windows client, a Collection Profile can be customized. A Custom Profile can include any combination of the predefined data categories, each with different collection interval frequencies.

Create a Custom Profile using IBM Systems Director Navigator for i

Use the following steps as guideline for creating a Custom Collection Profile:

1. In the configure Collection Services window, click the **Data to Collect** tab and select **Customize collection profile**.

2. Select a category from the **Available categories** section and click **Add** to add it to the Categories to Collect section, as shown in Figure 4-6.

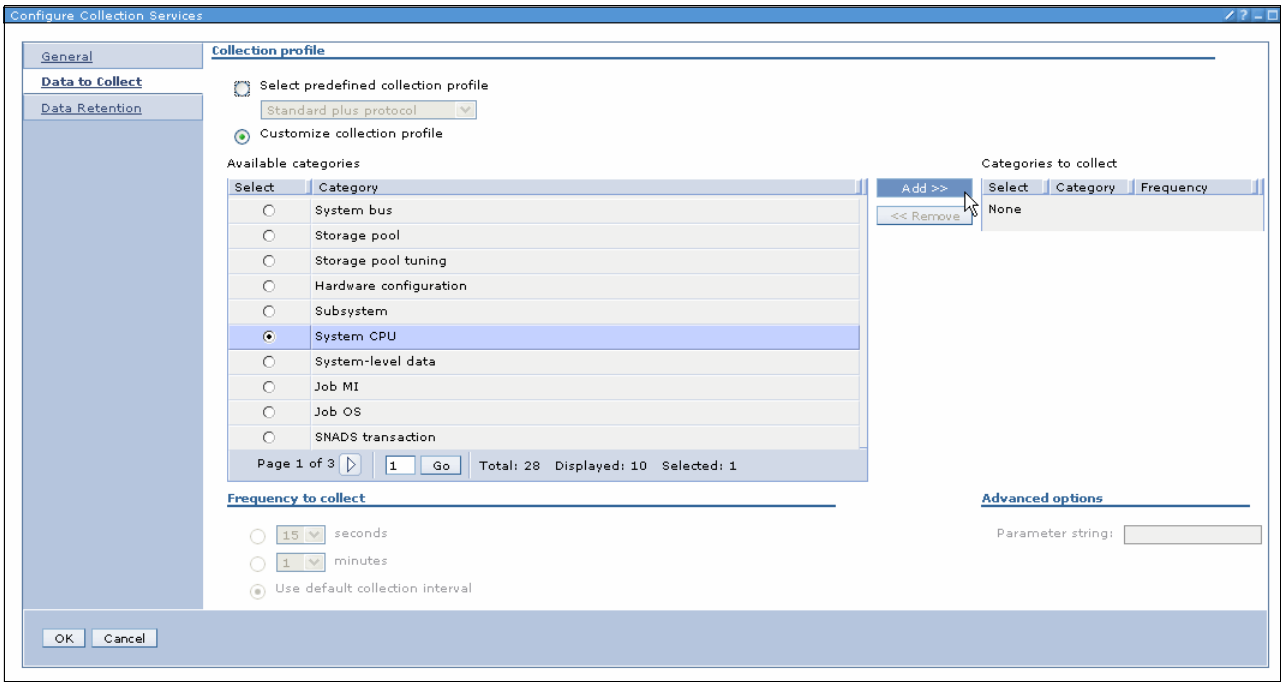


Figure 4-6 Collection profile categories

3. For a selected category in the right pane under Categories to collect, choose a frequency of data collection or accept the default collection interval for that category.
4. If there are additional options or parameters available for a category, they are shown under Advanced options. For example, if the **Local Response time** data category is selected, the response time buckets can be defined under Advanced options using different values for transaction times.
5. Repeat steps 2 to 4 for each data collection category in the custom profile.

Advanced customization options

Collection Services allows for extensive customization and enables applications to integrate performance data collection and define transactions specific to various applications.

User-defined categories

Collection Services can collect application specific data and store it in the Collection Object and make it available for real-time analysis using the Collection Object APIs.

Here are high level steps to implement the user-defined categories:

1. Create a program to collect performance data for the new category.
2. Register the new category and the associated program with Collection Services.
3. Add the new category to a Collection Profile and cycle Collection Services.
4. Develop a program to retrieve the data for analysis.
5. Optionally, create a data queue and register with Collection Services for interval and cycling notifications to enable real-time analysis.

For more information about implementing user-defined categories and program examples, go to the IBM i V6.1 Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Collection Services** → **Implementing user-defined categories in Collection Services**.

User-defined transactions

Collection Services provides the User Transactions data category and can collect transaction response time and other incremental metrics for a transaction, such as number of SQL statements used and number of programs called for the transaction. There are APIs you use to instrument your application to identify the start and end of a transaction. You need to update your application with these API invocation if you want to identify your transaction.

Once the application is instrumented to collect transaction data, Collection Services will begin collecting the performance metrics for the transactions at every interval.

The CRTPFRDTA command exports and organizes the transaction data into the QAPMUSRTNS file. The file contains one record per transaction type per job per interval. The data reported in the file includes standard data, such as the total transaction time and total number of transactions, as well as optional application-provided counters.

Collection Services summarizes transaction data within a job by transaction type and it stores transaction data for the first 15 transaction types; the rest are combined into the *OTHER transaction type.

> **Tip:** Collection Services does not collect system performance counters for the User Transactions(*USRTNS) category. However, the data in QAPMUSRTNS can be joined with the data in other performance database files to calculate and produce average resource consumption for different transaction types.

For more information about user-defined transactions and the related APIs, go to the IBM i V6.1 Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Collection Services** → **User-defined transactions**.

Web application performance

As briefly mentioned in 3.1.1, “Built-in metrics” on page 30, Collection Services can collect performance metrics for WebSphere Application Server jobs. Much of the data comes from WebSphere Performance Monitoring Infrastructure (PMI) data and transaction counters.

PMI is the core monitoring infrastructure for WebSphere Application Server and WebSphere family products like Portal, Commerce, and so on. The performance data provided by WebSphere PMI helps to monitor and tune the application server performance.

By default, Collection Services does not collect WebSphere Application Server performance data. It must be explicitly enabled by using the IBM Web Administration for i console. This can be done by performing these steps:

1. Open the administrative console.
2. Select **Servers** → **Application Servers** in the console navigation tree.
3. Click a server.
4. Click the **Configuration** tab.
5. Click **Performance Monitoring Infrastructure** (PMI) under Performance.
6. Select the **Enable Performance Monitoring Infrastructure (PMI)** check box.

7. Optionally, select the **Use sequential counter updates to enable precise statistic update** check box.
8. Optionally, choose a statistic set that needs to be monitored under Currently Monitored Statistic Set.
9. Optionally, click **Custom** to selectively enable or disable statistics. Choose a component from the left side tree and enable or disable statistics on the right side table. Go back to the main PMI configuration page by clicking the **Performance Monitoring Infrastructure** link.
10. Click **Apply** or **OK**.
11. Click **Save**.
12. Restart the application server. The changes you make will not take effect until you restart the application server.

Collecting data for an entire system with multiple LPARs

Collection Services can collect performance data for all logical partitions (LPARs) on a single server. It collects the performance data known to the Hypervisor and stores the data in the QAPMLPARH performance database file. The Hypervisor based LPAR data in QAPMLPARH, combined with System Processor data in QAPMSYSPRC, provides data on physical system resources and their utilization. This data can be analyzed by selecting **Performance Data Investigator** → **Collection Services** → **Physical System**.

For more information about this topic, refer to 8.8, “Collecting Collection Services data for an entire server” on page 232.

4.2.4 Managing Collection Services data

Collection Services stores all the data in management collection objects (*MGTCOL) to ensure the least amount of impact to the system during the collection process. Management Collection objects are then processed with the CRTPFRDTA command to generate the performance database files. For more information about these data files, go to the IBM i V6.1 Information Center and select **Systems management** → **Performance** → **Reference information for Performance** → **Collection Services data files**.

Management Collection objects can be managed separately and a different data retention policy can be applied than the one used for performance database files. This method also offers the flexibility to generate performance database files for a specific category and different interval levels than the one used in the management collection object. This allows for a drill-down type of analysis on a performance issue.

For example, Collection Services could be configured to collect data for all categories or a profile such as Standard Plus Protocol, with the collection interval set to 5 minutes and cycling every 24 hours. One set of performance database files could be generated at 15 minute intervals and can be used to run daily reports. Then, to analyze a particular performance issue, another set of performance database files could be generated at 5 minute intervals with only a single data category at a more granular level.

In general, IBM recommends always running Collection Services. Depending on the Data Retention attributes used when starting Collection Services, both the management collection objects and the performance database files in the *current* configured library are automatically deleted by Collection Services when the collections are cycled.

Collection Services data can also be moved between different systems and also be stored on a central system that is used for analysis. For special considerations when moving data from i5/OS V5R3 or IBM i V5.4 to an IBM i V6.1 system, refer to “Convert Data” on page 77.

Refer to 8.7, “Retaining collected data” on page 231 for some guidelines about Collection Services data retention.

4.3 Job Watcher

Job Watcher provides you with the ability to collect many different types of information about a set of jobs (or all jobs and tasks) on a system *and* it does this while providing near real-time, detailed, and summarized views of job, thread, and task performance data that enable you to begin your analysis immediately. The data is collected by a server job, stored in database files, and displayed on a client by way of the iDoctor GUI presentation facilities or in the browser by using the IBM Systems Director Navigator for i Performance interface.

4.3.1 How Job Watcher works

Job Watcher is similar to the sampling function of the system commands WRKACTJOB and WRKSYSACT in that each refresh computes delta information for the ending Snapshot™ interval. Refreshes can be set to occur automatically and as frequently as every 100 milliseconds. The data harvested from the jobs/threads/tasks being watched is done so in a non-intrusive manner (similar to WRKSYSACT).

The information harvested by Job Watcher includes:

- Standard WRKSYSACT type information:
  - CPU
  - DASD I/O breakdown
  - DASD space consumption
  - Current user profile
- Expanded details about types of waits and object lock/seize conditions
- Last run SQL statements syntax
- Program/procedure call stack, 1000 levels deep
- Job Watcher also includes some data not available anywhere else in *real time*:
  - Contention time (this includes objects being locked at the operating system level and objects being seized at the microcode level in addition to the application level object and record locks):
    - Holder and waiter job and thread
  - A breakdown of all other waits that occurred:
    - Details about the current wait
    - Duration of the wait
    - Object being waited for
    - Conflicting job/thread/task information
    - Specific LIC block point ID
- SQL statements, host variables, prepared statement arrays, and open cursors

- Communications data
- JVM statistics
- Activation group statistics

Similar to the other collectors, Job Watcher also collects wait information, but the biggest advantage of using Job Watcher for performance analysis is its ability to use a combination of wait accounting, call stack information, and conflicting job/thread/task information during any contention. The buckets in Job Watcher consist of many waits that are generally considered good or bad, and seeing the bad ones on a graph like Database Record Lock Contention makes it easy to identify problem areas for further investigation. Refer to 5.3.2, “Using Job Watcher for further investigation” on page 108 for an example.

When to run Job Watcher

Run Job Watcher when your system is running normally and you are not having performance problems. This gives you a health check of your system so that when it does have performance problems, you have a baseline to which you can make comparisons.

You might want to run Job Watcher over your system for a period in the morning, afternoon, and possibly during the evening when you might have more batch-type jobs running.

You can also run Job Watcher when your system is not running normally to try to identify the problems.

> **Tip:** You could also run Job Watcher for specific key jobs on your system when they are running well for baseline-comparison purposes. You will also want to run Job Watcher when you have an identified performance concern (with Collection Services data, for example) and need more detailed performance data for doing analysis and diagnostics.

4.3.2 Configuring Job Watcher

Job Watcher can be configured and started using either IBM Systems Director Navigator for i, IBM iDoctor, or Job Watcher CL commands.

Configure IBM Systems Director Navigator for i

Job Watcher can be configured using the Performance interface in IBM Systems Director Navigator for i by performing the following steps:

1. Go to Performance interface on the IBM Systems Director Navigator for i.
2. In the right pane, click **Show all Performance Tasks** to display all available Performance Management tasks.
3. Under the Performance Tasks, expand **Collectors** and select **Job Watcher** → **Add a Job Watcher Definition**.

> **Note:** The Job Watcher functions in the Director Navigator Performance interface are only displayed if the Performance Tools Job Watcher option is installed.

4. Make the changes as appropriate using the Add Job Watcher Definition wizard and verify that all the chosen options are on the Summary page, as shown on Figure 4-7.

Figure 4-7 Add Job Watcher definition

5. Click **Finish** to create the definition.

Starting a Job Watcher collection

Job Watcher can be started using the Performance interface in IBM Systems Director Navigator for i by performing the following steps:

1. Go to Performance interface on the IBM Systems Director Navigator for i.
2. in the right pane, click **Show all Performance Tasks** to display all available Performance Management tasks.
3. Under Performance Tasks, expand **Collectors** and select **Job Watcher** → **Start Job Watcher**.
4. Select a definition from the IBM-supplied or User-defined definition type lists and click **Next**.
5. Specify new Collection interval or other Termination options to override values specified in the Job Watcher Definition and click **Next**.

6. Verify the settings on the Summary page, as shown on Figure 4-8, and click **Finish** to start the Job Watcher.

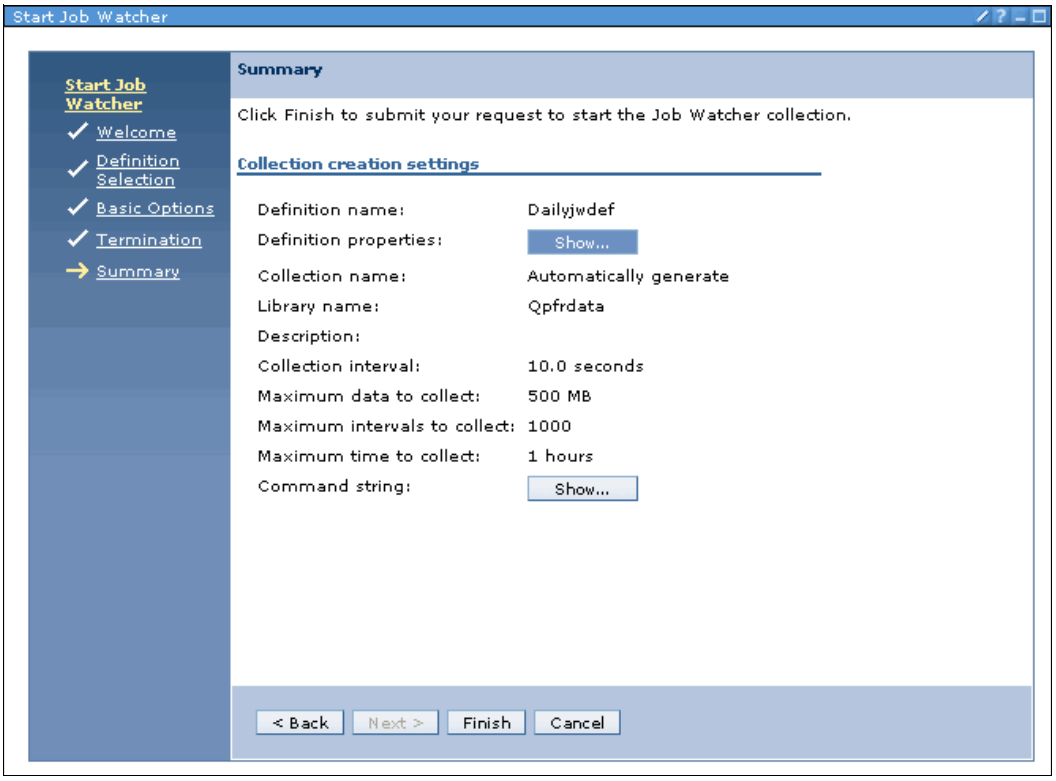


Figure 4-8 Start Job Watcher

For details about using the iDoctor Job Watcher, which includes topics about Monitoring and Analysis using Job Watcher, refer to *IBM iDoctor iSeries Job Watcher: Advanced Performance Tool*, SG24-6474, which can be found on the Web at the following address:

http://www.redbooks.ibm.com/abstracts/sg246474.html

4.4 Disk Watcher

Disk Watcher provides you with the ability to collect either a statistical summary of disk performance data or a trace of all disk I/O events that occur on a system for near real-time analysis.

Disk Watcher goes beyond the capabilities of the WRKDSKSTS and WRKSYSSTS commands that provide information regarding disk I/O, paging rates, CPU usage, and temporary storage. It provides detailed information related to Task Dispatching Elements(TDEs) and I/O queuing, objects, and programs involved in the I/O operation.

4.4.1 How Disk Watcher works

Disk Watcher collects detailed data about counts and durations of I/O operations to disk units. In addition, it provides useful runtime data regarding which objects are being read from disk or written to disk, and which jobs, threads, or system tasks are causing the operations. Figure 4-9 illustrates a high level overview of Disk Watcher functions.

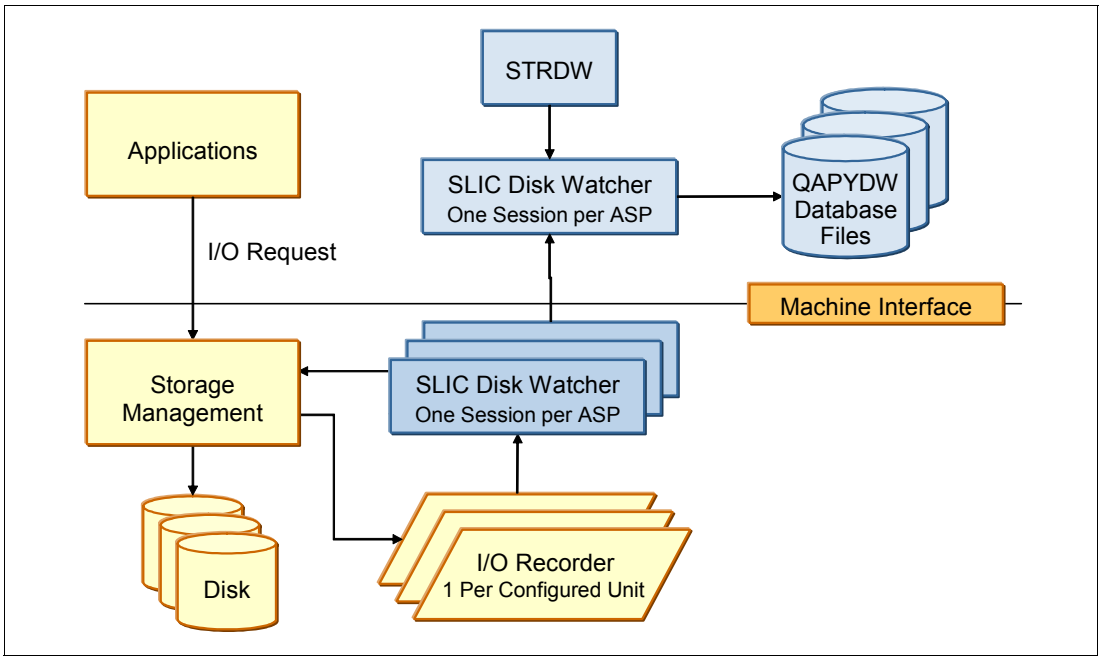


Figure 4-9 Disk Watcher high level function overview

4.4.2 Configuring Disk Watcher

Disk Watcher can be configured and started using either IBM Systems Director Navigator for i, IBM iDoctor, or Disk Watcher CL commands.

Configuration options

Disk Watcher configuration options can be divided into three main categories; options in each category are explained in the following sections.

Basic Disk Watcher definition options

Here is a list of basic Disk Watcher definition options:

Collection Type — Specifies the type of disk I/O collection to be performed, which determines the level detail of the data.

- **Statistical** — I/O data is summarized into a set of counters based upon the type of I/O operation for all I/O operations on the system.
- **Trace** — I/O data is gathered for every I/O operation that occurred on the specified disk or memory pool.
- **Full** — Both Statistical and Trace data are included.

Collection Interval — This is the interval between retrieval of disk I/O data. You can specify time or use dynamic intervals.

Disk Watcher definition advanced options

Here is the list of additional Disk Watcher advanced options:

Data availability	Specifies when the data is written to QAPYDW database files.
Collection file disk pool threshold	Species the percentage of disk pool that can be used before the collection will end. This threshold is for the auxiliary storage pool that contains the Disk Watcher database files. If the value specified is less than the current space available in the disk pool, the collection will end immediately.
System disk pool threshold	Specifies the percentage of the system disk pool that can be used before the collection is forced to end. Because Disk Watcher allocates temporary storage and the amount of data collected can be very large, this parameter allows you to limit how much of the system disk pool is consumed.

If the collection file exists in the system disk pool and values are specified in both “Collection file disk pool threshold” and “System disk pool threshold”, the value specified in “System disk pool threshold” takes precedence.

Note: If *SYSTEM special value is specified for the disk pool thresholds, Disk Watcher uses the percentage that is configured with the Change Storage Threshold function in Start System Service Tools (STRSST) command.

Disk and memory pool selection

Here are additional Disk Watcher selections to specify options for collecting disk I/O information:

Disk pools	Specifies the disk pools for which disk I/O information is collected.
Disk units	Specifies the disk units within the selected disk pools for which disk I/O information is collected.
Memory pools	Specifies the main memory storage pools for which disk I/O information is collected (as data is read from disk into memory or written from memory back to disk).

Object and program options

The Disk Watcher option that controls whether object and program information is collected when running in Trace mode is Object and program information. It specifies that object and program information, including their names, is collected. It also specifies whether this data collection will be limited to objects and programs that are resident in main storage. If you want to view the trace information from within the Systems Director Navigator Web Console, you need to specify *RESIDENT or *ALL for this parameter.

Note: Collecting information for objects that are not in main storage will take additional resources (CPU and I/O).

Conditional Collection options

Use the Conditional Collection options to specify additional conditional filters:

Collection Condition Specifies if the data collected should be limited to I/O operations whose Response time, Wait time, or Service time meet a specified condition.

Collection Range Specifies if the data collected should be limited to I/O operations whose Response time, Wait time or Service time fall in a specified range.

Add a definition using IBM Systems Director Navigator for i

Disk Watcher can be configured using the Performance interface in IBM Systems Director Navigator for i by performing the following steps:

1. Go to the Performance interface on the IBM Systems Director Navigator for i.
2. in the right pane, click **Show all Performance Tasks** to display all available Performance Management tasks.
3. Under the Performance Tasks, expand **Collectors** and select **Disk Watcher** → **Add a Disk Watcher Definition**.
4. Make the changes as appropriate using the Add Disk Watcher Definition wizard and verify all the chosen options on the Summary page, as shown in Figure 4-10.

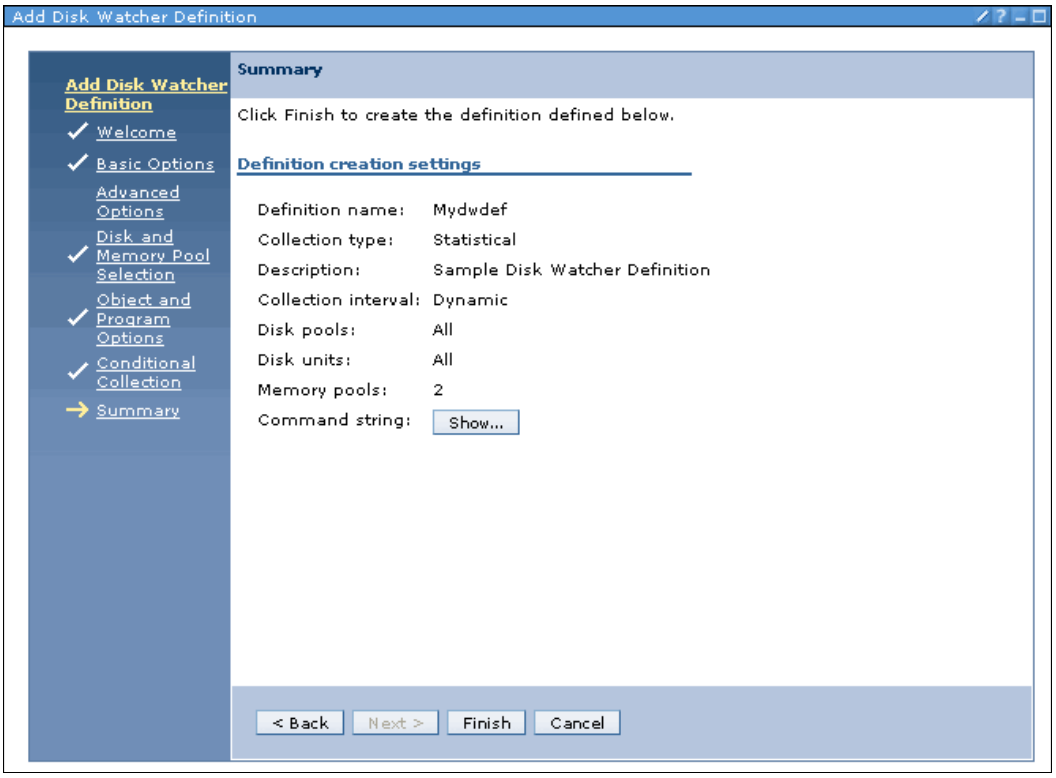


Figure 4-10 Add Disk Watcher Definition

5. Click **Finish** to create the definition.

Note: The Disk Watcher GUI options are only available if the Performance Tools License program manager feature is installed.

Start Disk Watcher using IBM Systems Director Navigator for i

Disk Watcher can be started using the Performance interface in IBM Systems Director Navigator for i by performing the following steps:

1. Go to Performance interface on the IBM Systems Director Navigator for i.
2. in the right pane, click **Show all Performance Tasks** to display all available Performance Management tasks.
3. Under the Performance Tasks, expand **Collectors** and select **Disk Watcher** → **Start Disk Watcher**.
4. Select a definition from the IBM-supplied or User-defined definition type list and click **Next**.
5. Specify a new Collection interval or other Termination options to override values specified in the Disk Watcher definition and click **Next**.
6. Verify the settings in **Summary** page, as shown on Figure 4-11, and click **Finish** to start the Disk Watcher.

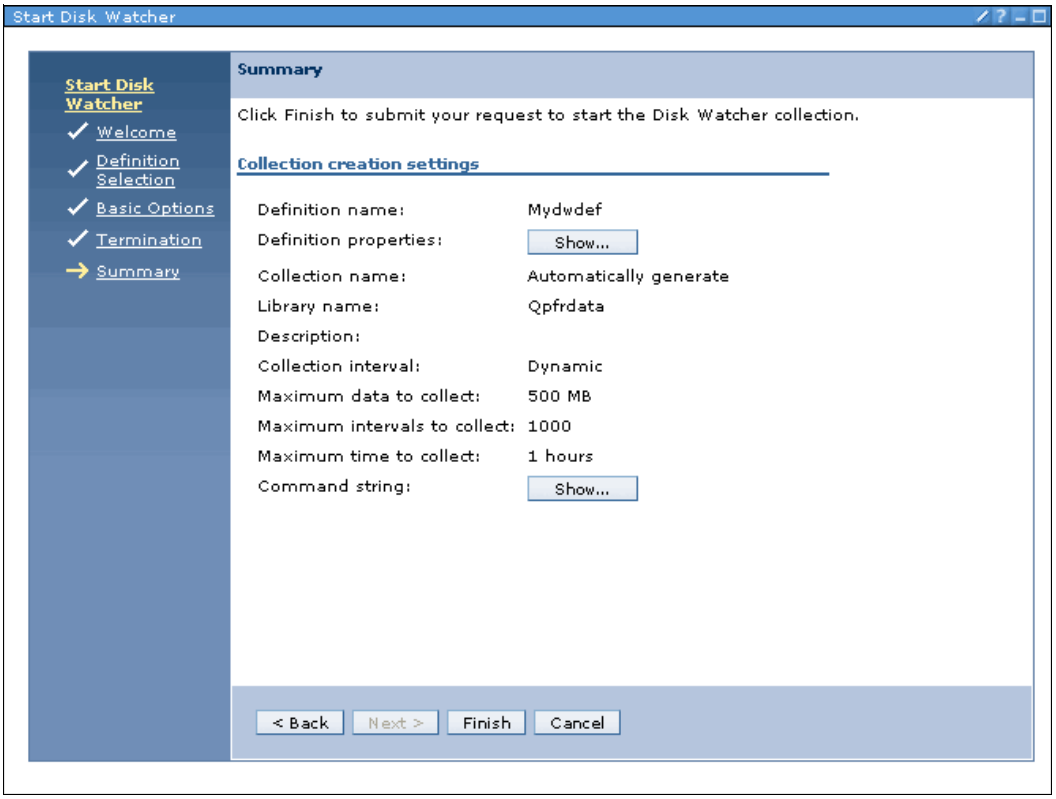


Figure 4-11 Start Disk Watcher

4.5 Performance Explorer

Performance Explorer (PEX) is a data collector that can be used to gather very detailed information about a specific application, program, or system resource.

Performance Explorer is similar to Collection Services in some aspects. They both collect data into a management collection object that is later placed into performance database files for analysis and reporting. In both tools, specific areas of interest can be specified using predefined profiles. However, Performance Explorer differs significantly from Collection

Services when it comes to the level of detail in which the data is collected. It is a tool that can be used for statistical analysis and profiling of specific system and application resources.

Collection Services collects a broad range of system data at regularly scheduled intervals, with minimal system resource consumption. In contrast, Performance Explorer starts a session that may collect a significant amount of data. This PEX collection may have a large amount of trace-level data. This trace generates a large amount of detailed information about the resources consumed by an application, job, or thread. Specifically, you can use Performance Explorer to answer specific questions about areas like system-generated disk I/O, procedure calls, page faults, and other trace events. It is the ability to collect very specific and very detailed information that makes the Performance Explorer effective in helping isolate performance problems. For example, Collection Services can tell you that disk storage space is rapidly being consumed. You can use Performance Explorer to identify what programs and objects are consuming too much disk space, and why.

When Performance Explorer is running, it creates only the files that are needed for the collection.

Note: You can collect Performance Explorer data and Collections Services data at the same time. Any mix of all these performance data collectors (CS, JW, DW, and PEX) can run at the same time.

4.5.1 How Performance Explorer works

The following general steps, along with the chart in Figure 4-12, illustrate the basic workflow in Performance Explorer:

1. Define a Performance Explorer Definition. PEX Definition tells the Performance Explorer what data to gather.
2. Start the Performance Explorer based on the definition.
3. Run the application or program that needs to be analyzed.
4. End the collection, which saves the collected to a set of database files.
5. Generate and print the analysis reports from the database files.

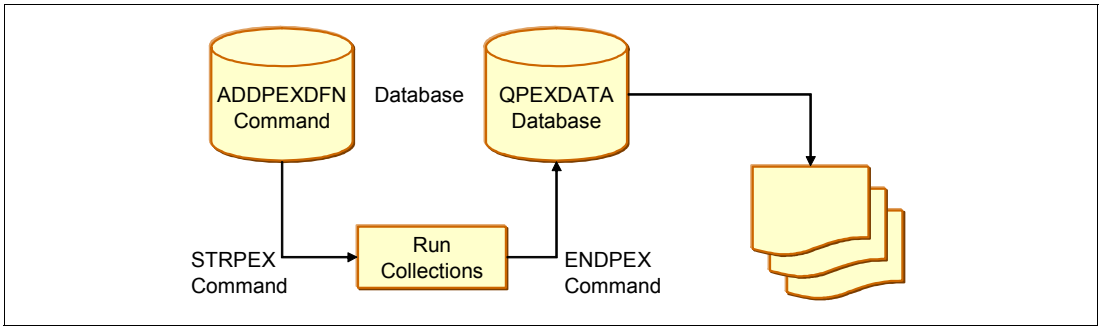


Figure 4-12 Performance Explorer workflow

4.5.2 Configuring Performance Explorer

Performance Explorer can be configured and started using either IBM iDoctor or Performance Explorer CL commands.

Performance Explorer definitions

The parameters and conditions that determine what data is collected and how it is collected are configured and stored using Performance Explorer definitions. A PEX definition is a member in a specific system database (QUSRSYS/QAPEXDFN), and it controls most of the aspects of making a PEX collection except for the following:

- When to begin making the collection.
- When to end the collection.
- The library where collection data is stored.

A PEX definition also controls:

- Which subset of the hundreds of event types are to be activated.
- Granularity of CPU sampling.
- Maximum amount of data to be collected.
- Subset of jobs or system tasks (or all jobs and all tasks) that are to be traced.

The PEX Definition Wizard within IBM iDoctor for IBM i PEX-Analyzer is a full-featured interface over the ADDPEXDFN IBM i command. Use the PEX Definition Wizard to create or modify a PEX definition (which is discussed within the remainder of this section).

PEX definition types

Here is a list of the three PEX definition types:

Collection type — Specifies the type of performance data to be collected.

- **Statistical** — General performance program statistics are collected to help identify problem areas. This mode is mainly used as a map to help determine if and where more detailed information should be collected and analyzed. The statistics data can be structured in two different ways:
  - **Flat** — A flattened structure organizes the statistics into a simple list of programs or procedures, each with its own set of statistics. The Performance Explorer tool will not collect data for a parent-child relationship.
  - **Hierarchical** — A hierarchical structure organizes the statistics into a call tree form in which each node in the tree represents a program procedure run by the job or task. The Performance Explorer tool will collect data for a parent-child relationship.
- **Trace** — Detailed trace information is collected. Gathers a historical trace of performance activity generated by one or more jobs on the system. The trace type gathers specific information about when and in what order events occurred. The trace type collects detailed reference information about programs, Licensed Internal Code (LIC) tasks, IBM i jobs, and object reference information. This is the most detailed type of performance data collection available.
- **Profile** — Specific programs are sampled to identify sections of code that use larger amounts of resources. PEX identifies high-level language (HLL) programs, modules, procedures, and statements that consume excessive CPU utilization based on source program statement numbers. This information is very valuable when you want to improve the performance of a specific program or application.

Configuring Performance Explorer using IBM i commands

To collect detailed trace information, you need to tailor Performance Explorer to work optimally with the application process from which the trace is being taken.

To configure Performance Explorer, perform these steps:

1. Create a session definition that informs the system which performance data you want to collect. On the Add Performance Explorer Definition (ADDPEXDFN) display, specify the collection type and a name for the definition. This definition is stored as a database member by that name in the QAPEXDFN file in library QUSRSYS. The name that you specify is used by the Start Performance Explorer (STRPEX) command.
2. (Optional) Add a filter (Add PEX Filter (ADDPEXFTR) command). A Performance Explorer filter identifies the performance data that is to be collected during a Performance Explorer session, and is meant to limit the amount of data collected by specifying a compare value for specific events.
3. Start collecting data (Start Performance Explorer (STRPEX) command). A job may be in more than one Performance Explorer collection if the *PMCO event is not being collected. If the *PMCO event is being collected, then a job can be in more than one collection only if all the collections have the same interval specification (ADDPEXDFN INTERVAL() parameter). You can specify a definition and optional filter in the STRPEX command.
4. Run your command, program, or workload for data that you want to analyze.
5. Stop collecting the data and save it to database files for analysis. Use the End Performance Explorer (ENDPEX) command to stop the collection.
6. Analyze the performance data. The Print Performance Explorer Report (PRTPEXRPT) command provides unique reports for each type of data (statistical, profile, trace profile, or trace).

   The following are other options for analysis:

   – Write your own queries for the set of database files.
   – Use IBM iDoctor for IBM i PEX-Analyzer. IBM i Doctor for IBM i is a set of software performance analysis tools and associated services that extend your ability to evaluate the health of your system by gathering detailed information and providing automated, graphical analysis of this data.

   Note: There is some limited support for PEX data on Performance Data Investigator (PDI).

All of the Performance Explorer commands can be accessed by using one of the following methods:

- The command interface. Type the commands from the command line. All the commands are part of the IBM i base operating system.
- The Performance Tools menu options.

For further information about Performance Explorer, refer to Chapter 11, “Performance Explorer”, in *Performance Tools for iSeries*, SC41-5340.

For further details about PEX Analyzer in IBM iDoctor, refer to *Application and Program Performance Analysis Using PEX Statistics on IBM i5/OS*, SG24-7457, which can be found at the following address:

http://www.redbooks.ibm.com/abstracts/sg247457.html

4.6 Managing collections

the Performance interface on IBM Systems Director Navigator for i allows you to manage all of your performance data in both file based as well as management collection objects on the entire system.

4.6.1 Working with the Collections interface

To open the Collections interface, go to the Performance tab on IBM Systems Director Navigator for i and click **Collections**. Alternatively, if you are in Show All Performance Tasks, you can simply click the context menu next to **Performance** and choose **Collections** in the tasks pane. A sample Collections window is shown in Figure 4-13.

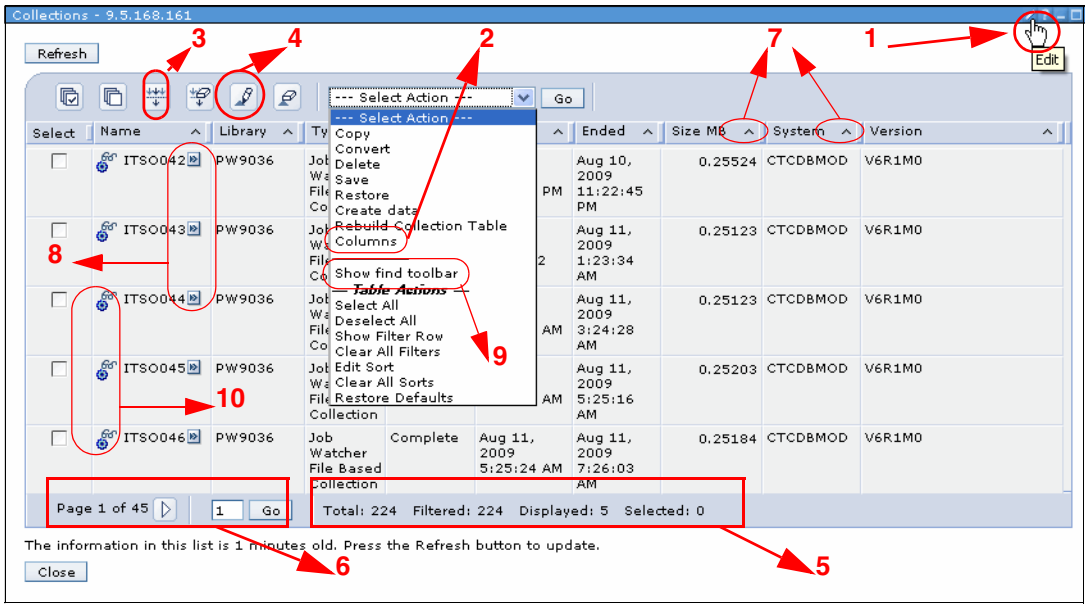


Figure 4-13 Working with Collections

The Collections interface functions similar to most other table windows within IBM Systems Director Navigator for i. The window customization options, navigation within the tables, and context menus are all consistent throughout the product.

Table properties

You can customize the window by changing the number of rows and columns displayed.

Number of Items in the table

For faster page loads or easier navigation, you may want to decrease or increase the number of items displayed within the table. The default value for this User Preference is 15 rows per page within the table window.

To bring up the User Preferences, click the Edit icon at the top right corner of the Collections window (shown by **1** in Figure 4-13).

The new value specified in User Preferences will take effect when you close and reopen the Collections window.

Refresh

The list of all collections on the system is maintained in the QAPMCCCNTB table within the QUSRSYS library so that the Collections task and Performance Data Investigator can display the entries quickly without having to search all the libraries on the entire system every time. The collectors and the related commands update this table whenever collections are created on the system. The **Refresh** button on the Collections window shown in Figure 4-14 will reload the table from QAPMCCCNTB.

Rebuild Collection Table

It may be necessary to rebuild the Collections repository that is used by the Performance tasks if the QAPMCCCNTB table is not updated properly. For example, if a collection was restored from another system using RSTOBJ or RSTLIB commands instead of the RSTPFRCOL command or if the collector job was ended abnormally either due to system or subsystem ending immediately. Also, when generic object functions such as DLTLIB command are used to delete performance collections, the collections repository in QUSRSYS/QAPMCCCNTB will become inconsistent and may not show the actual collections that exist on the system. Choosing the **Rebuild Collection Table** option from the Actions drop-down menu in the Collections window shown in Figure 4-14 will search the entire system for collections and rebuild the collections repository.

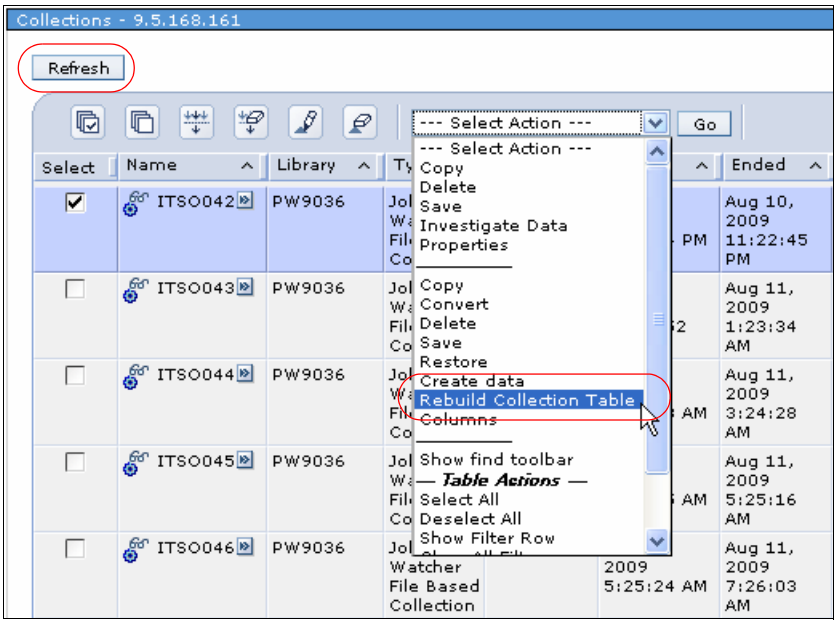


Figure 4-14 Refresh and Rebuild Collection Table options

Columns in the table

Use the Columns window to specify which columns of information are displayed in the table.

You can open the Columns window by selecting **Columns** from the Select Action drop-down menu, as shown by **2** on Figure 4-13 on page 70, and clicking **Go**.

Displayed columns are listed under Current columns. Additional columns are listed under Available columns, as shown in Figure 4-15.

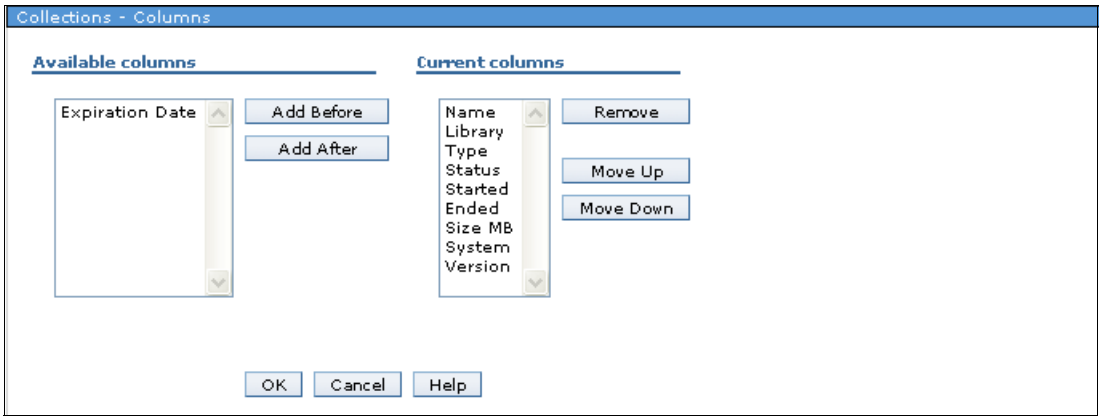


Figure 4-15 Collections: Columns menu

In the Available columns list, select any columns you want displayed and use the **Add Before** and **Add After** buttons to add them to the Current columns list in the order you want them displayed. To remove or reorder columns from the Current columns list, select the column and click **Remove**.

When you click **OK**, you will return to the table and the columns displayed will be updated to reflect your customizations.

Filter

The filtering feature is very powerful and versatile and it allows you to define custom filters for multiple columns and the filter criteria is column data sensitive. This powerful filtering function:

- Supports filtering on one or more columns in the table.
- Supports the following filter conditions for text columns:
  - Contains.
  - Starts with.
  - Ends with.
  - *Match case* option can be applied to all three conditions above.
- Supports the following filter conditions for numerical columns:
  - All numbers.
  - Numbers less than.
  - Numbers less than or equal to.
  - Numbers greater than.
  - Numbers greater than or equal to.
  - Numbers equal to.
  - Numbers not equal to.
  - Numbers between.
  - Numbers between and including.
- Supports the following filter conditions for Date columns:
  - All Dates.
  - Dates until.
  - Dates from.
  - Dates between.

You can define filtering criteria after displaying the filter row by clicking the **Show Filter Row** toggle icon shown by **3** on Figure 4-13 on page 70.

You can click the same icon to toggle between show and hide for the filter row. When the filters are defined for multiple columns, you can toggle individual filters by selecting the check box to make it Active or Inactive. You can clear all filters by clicking the **Clear All Filters** icon.

> **Note:** Hiding the filter row does not affect the filter criteria that is already active. You can check the status area shown by **5** on Figure 4-13 on page 70 to determine if any filters are active.

Sort

The built-in data sorting function allows you to:

- Specify up to three columns from the table to sort the table by.
- Specify ascending or descending sorting for each of the columns.

Click the **Edit Sort** icon shown by **4** on Figure 4-13 on page 70 to define how you want to sort the table items. You can also simply click the caret symbol to sort a single column, as shown by **7** on Figure 4-13 on page 70.

When sorting is defined for multiple columns, they are indicated by a numbered icon that shows the order of the sort under the respective columns.

Selection and Navigation

You can perform actions on multiple items in the table if you select them by clicking the check box under the **Select** column. When more than one item is selected, you will be able to copy, delete, save, and so on using the Select Action drop-down menu. Additionally, when there are no selections in the table or when you select a single item from the table, you will be able to either investigate or convert data.

Find

The Find feature is another very useful feature, especially when you are working on a system with several hundred performance data collections. You can quickly find the collection(s) you want to work with by searching for a word in one or more columns.

To bring up the Find toolbar, select **Show find toolbar** in the drop-down menu, as shown by **9** in Figure 4-13 on page 70 and click **Go**. Using the Find function, you can:

- Specify four different conditions:
  - Contains.
  - Starts with.
  - Ends with.
  - Exact match.
- Limit the search to:
  - A specific column.
  - Search across all columns.
- Search up or down in the table.
- Specify whether to *Match case* or not.

In the Find toolbar, enter a text string to look for and define the other search criteria and click **Find**. The function will then search for a match and displays the first matching item in the table and the item will be selected and highlighted for further actions. Clicking **Find** again will take you to the next match in the table.

4.6.2 Working with collections

The Collections interface under the Performance tab in IBM Systems Director Navigator for i provides a central place to manage and work with all of the performance data that is on the system. It shows performance data collections that are collected using Collections Services, Job Watcher, Disk Watcher, or Performance Explorer. The type of the collection can be easily identified by referring to the icons next to them, as shown by **10** on Figure 4-13 on page 70 or by looking at the text under the Type column.

Actions on a collection

Using the Collections interface, actions can be performed on either individual collections or group of items. Depending on the selections made in the table, the available actions are shown in the drop-down menu, as shown in Figure 4-16.

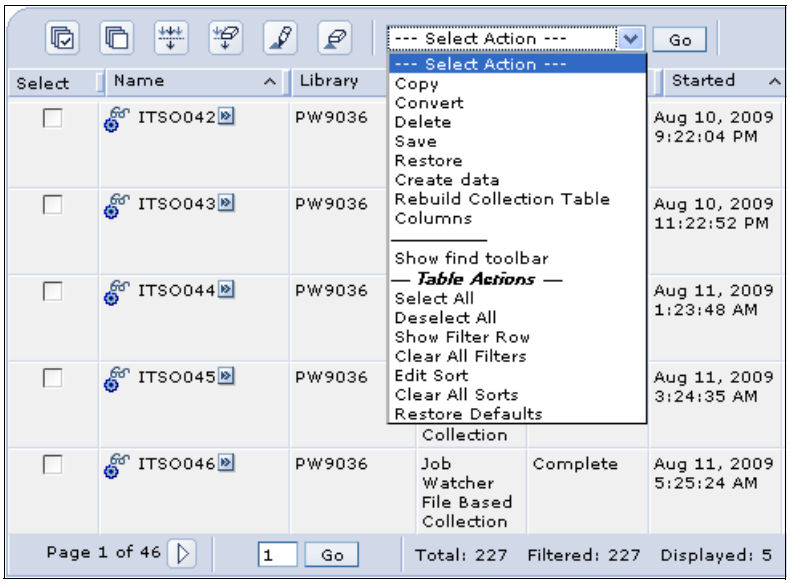


Figure 4-16 Select Action drop-down menu

Actions on individual collections can also be performed using the context menu of the corresponding collection, as shown in Figure 4-17.

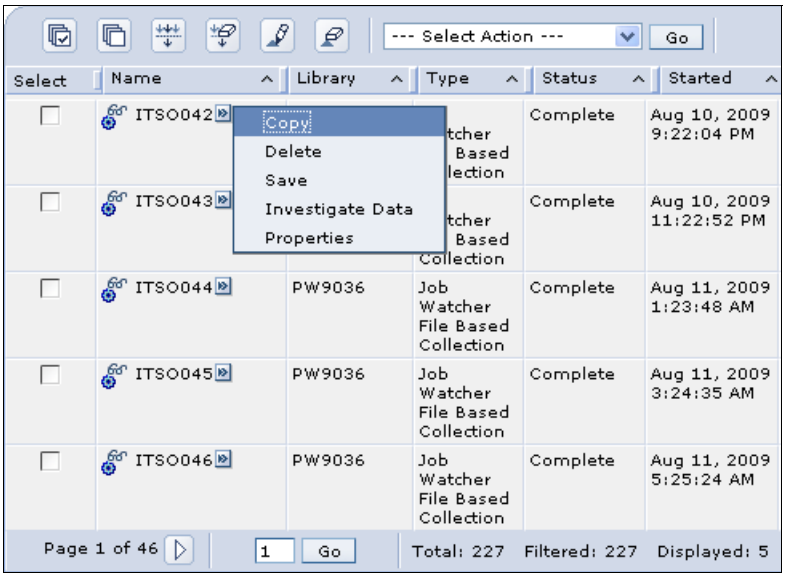


Figure 4-17 Collection context menu

Note: Be careful not to select too many collections when performing actions, because the action will be performed on each of the selected items one by one, and it may take a long time to cycle through all of the items.

Copy Collection

One or more collections can be copied to another library or to another collection using the Copy Collection function. A sample Copy Collection page is shown in Figure 4-18.

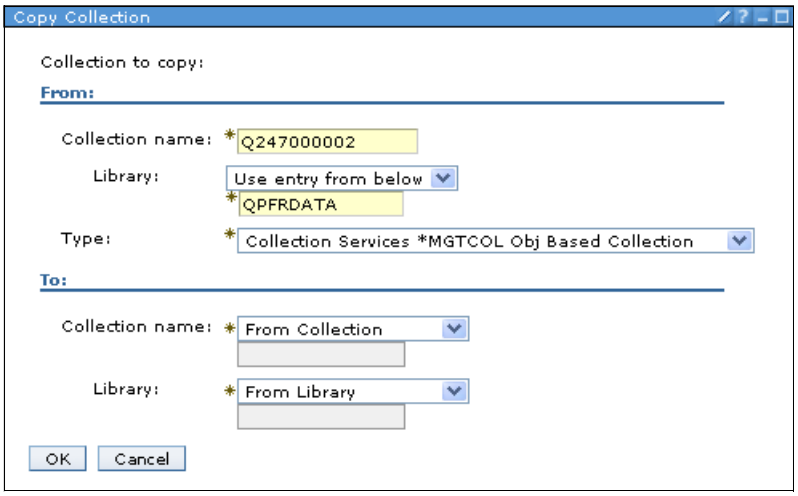


Figure 4-18 Copy Collection

Delete Collection

One or more collections can be deleted using the Delete action. The selected collections are displayed in Delete Collection window, as shown in Figure 4-19.

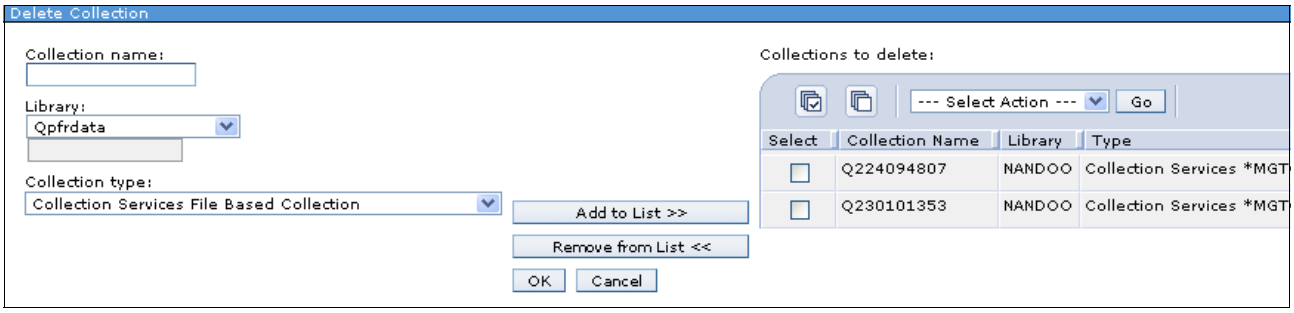


Figure 4-19 Delete Collection window

If there are any failures during the delete operation, a message window will open and you can review the details. A sample status window is shown in Figure 4-20.

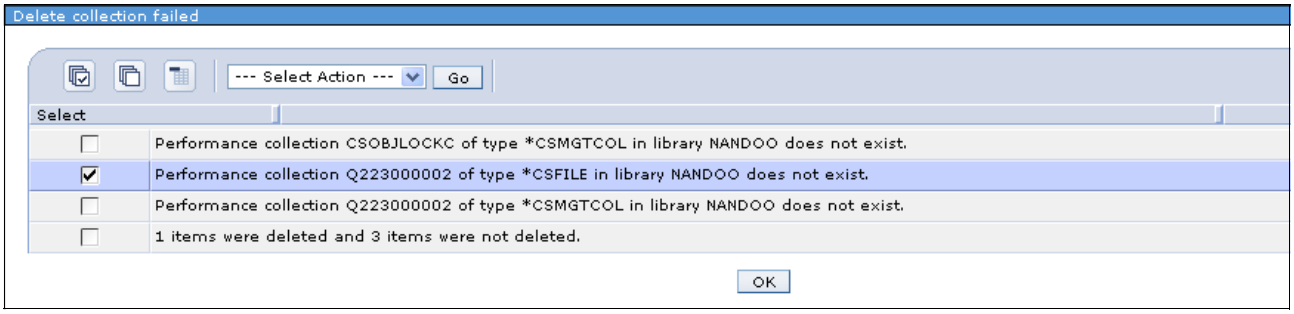


Figure 4-20 Delete collection failed

Save

When you select **Save** from the Actions drop-down menu, the Save Collection page is displayed, which allows you to select one or more collections to save. Click **Add to List** to add the specified collection to the list of collections to save. Click **Remove from List** to remove the specified collection from the list of collections to save.

> **Note:** All of the collections to be saved must be in the same library.

Investigate Data

For a single collection, you can select the **Investigate Data** option from either the context menu or the Action drop-down menu to open the default perspective in the Investigate Data window for that particular collection type. For Collection Services and Job Watcher collections, the default perspective is Resource Utilization Overview. For Disk Watcher collections, the default perspective in the Investigate Data window is Disk Statistics Overview.

> **Note:** The Investigate Data option is available only for file based collections. To analyze Collection Services data using the Investigate Data function, performance database files must be created using the CRTPFRDTA command if the Create database files option is set to *NO when Collection Services is configured.

Chapter 5, “Performance data analysis” on page 81 discusses the Investigate Data option in greater detail.

Create Data

In the Collections window, the Create Data option is available in the context menu for a Collection Services *MGTCOL based collection. When this option is chosen, the Create Collection page opens, which allows you to create database files based on a management collection object (MGTCOL). A sample Create Collection window is shown in Figure 4-21.

When creating the file based collection, the sampling interval, range of data, and the categories of data do not need to be the same as the actual configured values in Collection Services. Refer to 4.2.4, “Managing Collection Services data” on page 58 for more information.

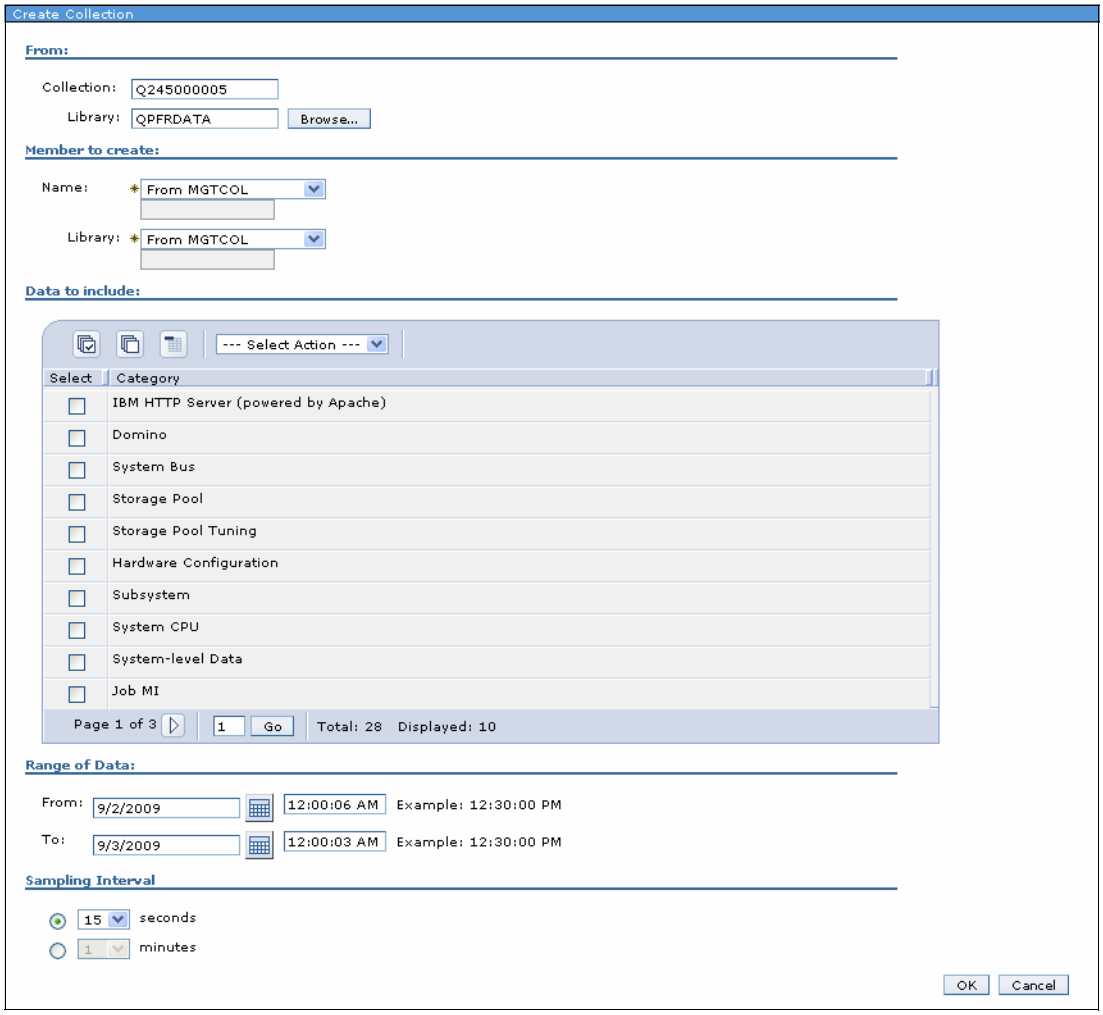


Figure 4-21 Create Collection

Convert Data

Collection Services and Disk Watcher data collected on i5/OS V5R3 or IBM i V5.4 can be viewed with Performance Data Investigator on V6.1. The steps and considerations for viewing performance data from i5/OS V5R3 or IBM i V5.4 releases follow.

Note: Job Watcher data collected on i5/OS V5R3 and IBM i V5.4 systems can only be analyzed using the iDoctor GUI client.

Note: Moving the management collection object is the preferred way to move CS data from one partition to another.

Collection Services

- When the performance data from i5/OS or IBM i V5.4 is in a Management Collection object:
  a. Use the SAVOBJ command to save the Management Collection object to a save file.
  b. FTP the saved file to the V6.1 system.
  c. Run the Restore Performance Collection (RSTPFRCOL) command or select the **Restore collection** option from the Actions list on Collections window in IBM Systems Director Navigator for i.
- When the performance data from i5/OS V5R3 or IBM i V5.4 is in Collection Services database files:
  a. Run the SAVOBJ command to save the Collection Services database files (QAPM*) to a save file.
  b. FTP the saved file to the V6.1 system.
  c. Run the Restore Performance Collection (RSTPFRCOL) command or select the **Restore collection** option from the Actions list on Collections window in IBM Systems Director Navigator for i.
  d. Run the Convert Performance Collection (CVTPFRCOL) command or select the **Convert Collection** option in Collections window on IBM Systems Director Navigator for i.

Note: When using iDoctor, we recommend not converting the data. In some instances, this will create problems when trying to view wait buckets that have changed from release to release.

Disk Watcher

Disk Watcher data collected on i5/OS V5R3 or IBM i 5.4 release can be viewed with the Performance Data Investigator by performing these steps:

1. Use SAVPFRCOL command to save the Disk Watcher database files (QAPYDW*) to a saved file.
2. FTP the save file to the V6.1 system.
3. Run the Restore Performance Collection (RSTPFRCOL) command or select the **Restore collection** option from the Actions list on Collections window in IBM Systems Director Navigator for i.
4. Run the Convert Performance Collection (CVTPFRCOL) command or select the Convert Collection option in Collections window on IBM Systems Director Navigator for i.

Note: Disk Watcher data collected on i5/OS V5R3 or IBM i V5.4 must include the QAPYDWHDWR file and this can be done by setting the environment variable QPY_DW_HDWR_OPT to 1 before starting the Disk Watcher collection. On V6.1, the Disk Watcher definition used must also include Object Information and this can be done using the OBJINF(*RESIDENT) or OBJINF(*ALL) option with the STRDW command.

Collection Properties

For file based collections, the Properties option shows general collection attributes, such as Collection Name, Library, Start and End Timestamps, Size, Collector, the name of the job that created the collection, and the definition used.

For Management Collection (*MGTCOL) based collections, the Properties option shows most of the general attributes as well as the configured attributes of the collection and its expiration date. Using the General tab in the Properties window, you can also change the expiration date or set it to never expire.

5

Performance data analysis

In this chapter, we describe how to navigate the Performance Data Investigator windows for Performance in IBM Systems Director Navigator for i. We then discuss how you can use the perspectives to analyze collected data. We walk you through a data analysis using real performance data and identifying real performance issues in the collected data. Where possible, we also provide suggestions about how the identified issues could possibly be addressed.

During the exercise, we uncover two key capabilities of the collector analysis tools:

- The ability to quickly drill down and find the root cause of a particular performance issue
- The ability to uncover unknown performance problems

© Copyright IBM Corp. 2009. All rights reserved.

5.1 Accessing the Investigate Data window

The Investigate Data windows can be accessed by way of the Performance link on the Welcome page of IBM Systems Director Navigator for i. To work with IBM Systems Director Navigator for i, access the following URL from a Web browser, where *hostA* is your IBM i name:

`http://hostA:2001`

> **Note:** The HTTP Administration Server must be active on IBM i to access IBM Systems Director Navigator for i. The IBM i default setting is to start the HTTP Administration Server automatically after an IPL.

You will be presented with the logon page, as shown in Figure 5-1.

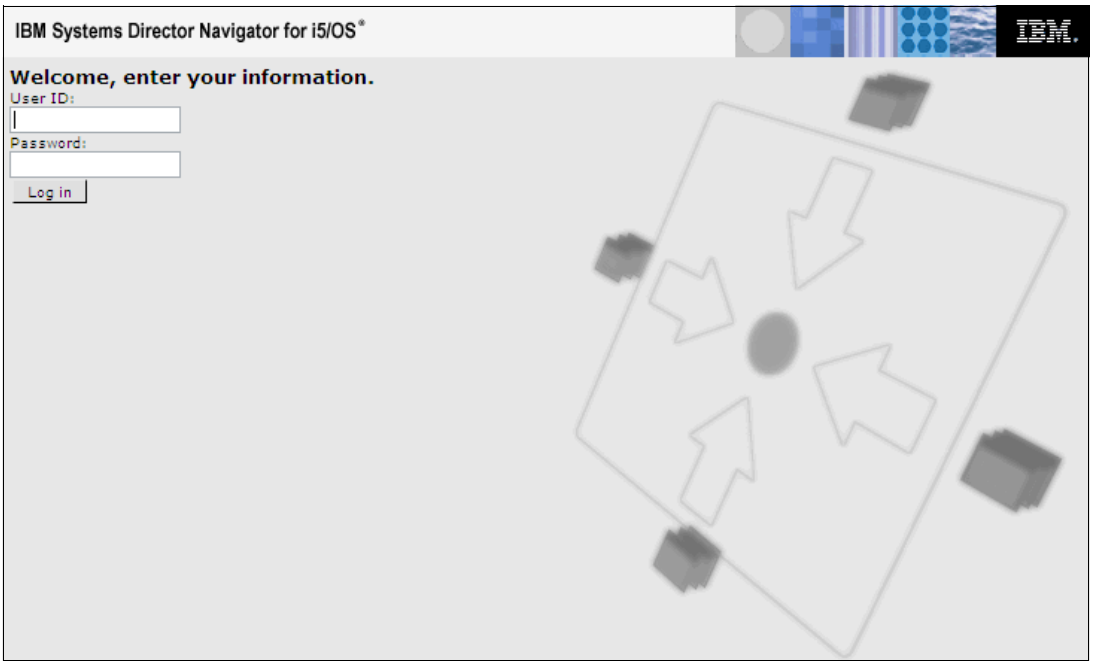


Figure 5-1 IBM Systems Director Navigator for i log in

Enter your user ID and password and click **Log in**. You will be presented with the Welcome page, where you can expand i5/OS Management, as shown in Figure 5-2 on page 83.

> **Note:** The IBM Systems Director Navigator for i5/OS was renamed IBM Systems Director Navigator for i in 2008. This occurred after the release of the product. This is why you will still see references to i5/OS in many of the screen captures in this IBM Redbooks publication.

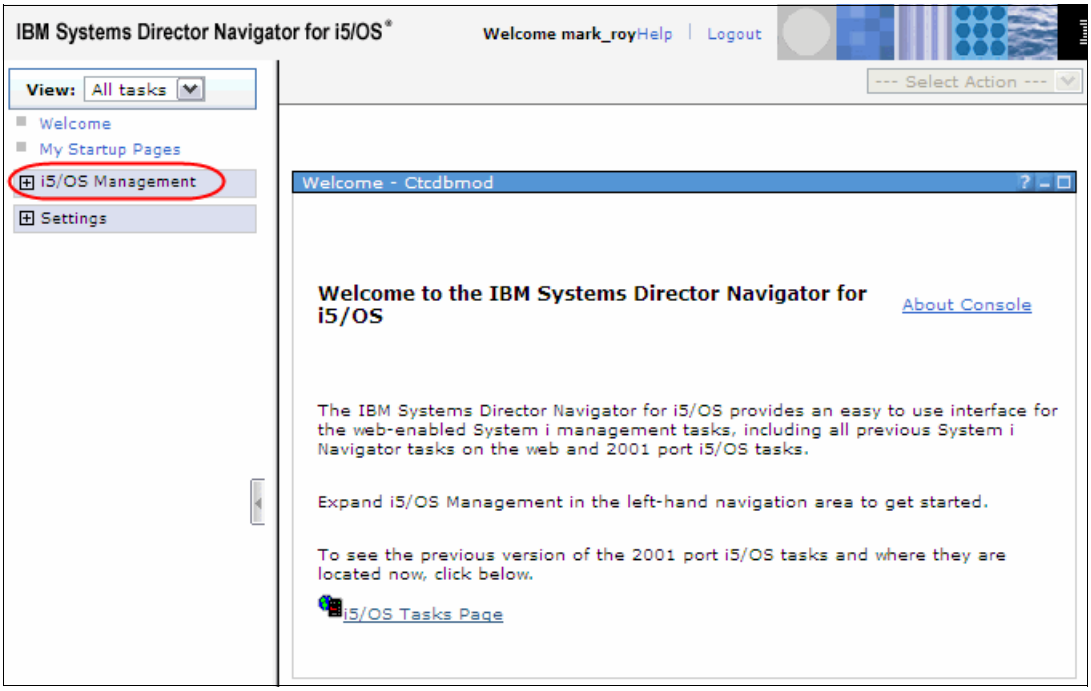


Figure 5-2 IBM Systems Director Navigator for i welcome page

Expand the **i5/OS Management** task and then select **Performance**, as shown in Figure 5-3.

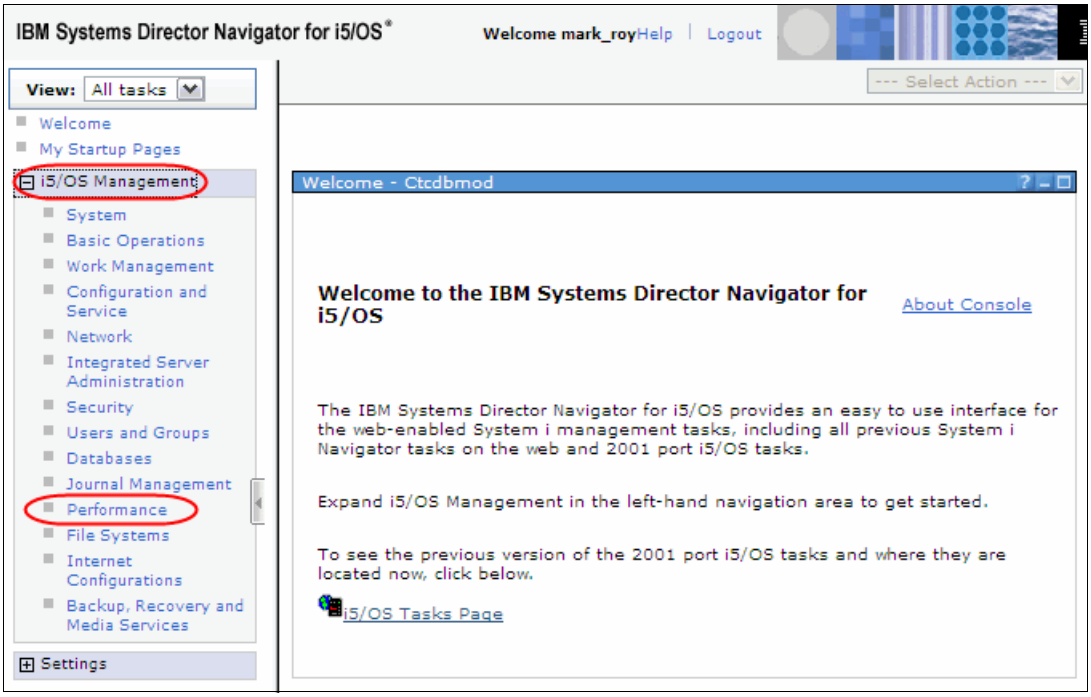


Figure 5-3 Selecting the Performance link

You will be presented with the Performance pane, where you can select **Investigate Data**, as shown in Figure 5-4.

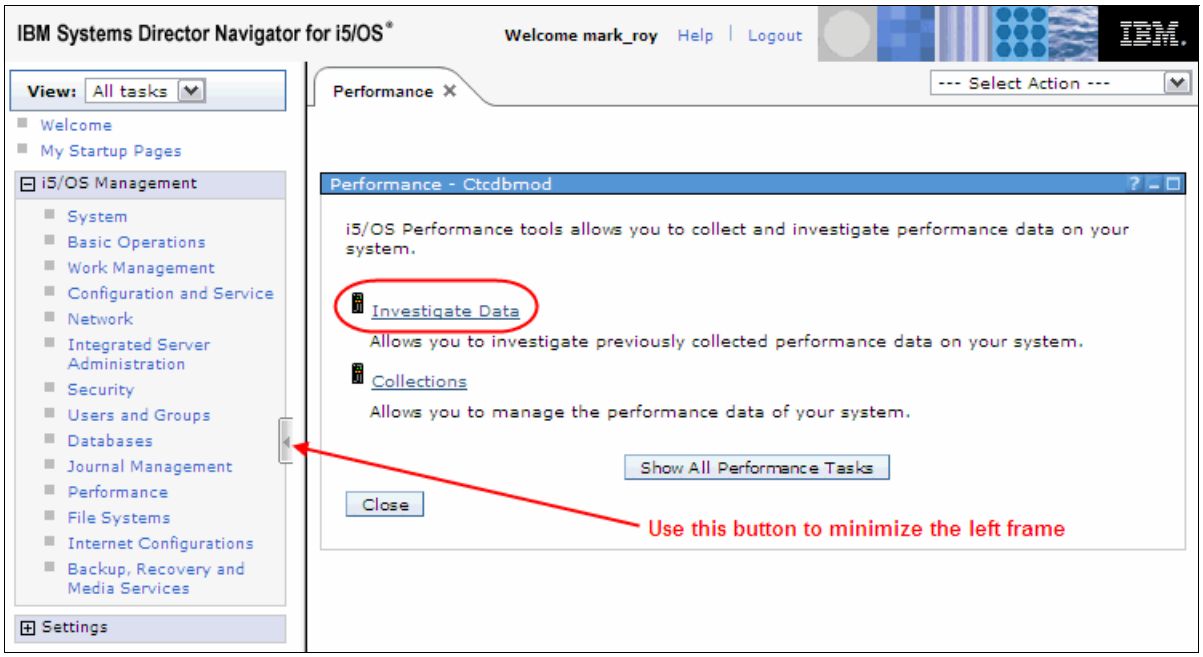


Figure 5-4 Selecting the Investigate Data link

5.2 Navigating around Investigate Data

When you select the **Investigate Data** link, the Investigate tab opens, as shown in Figure 5-5.

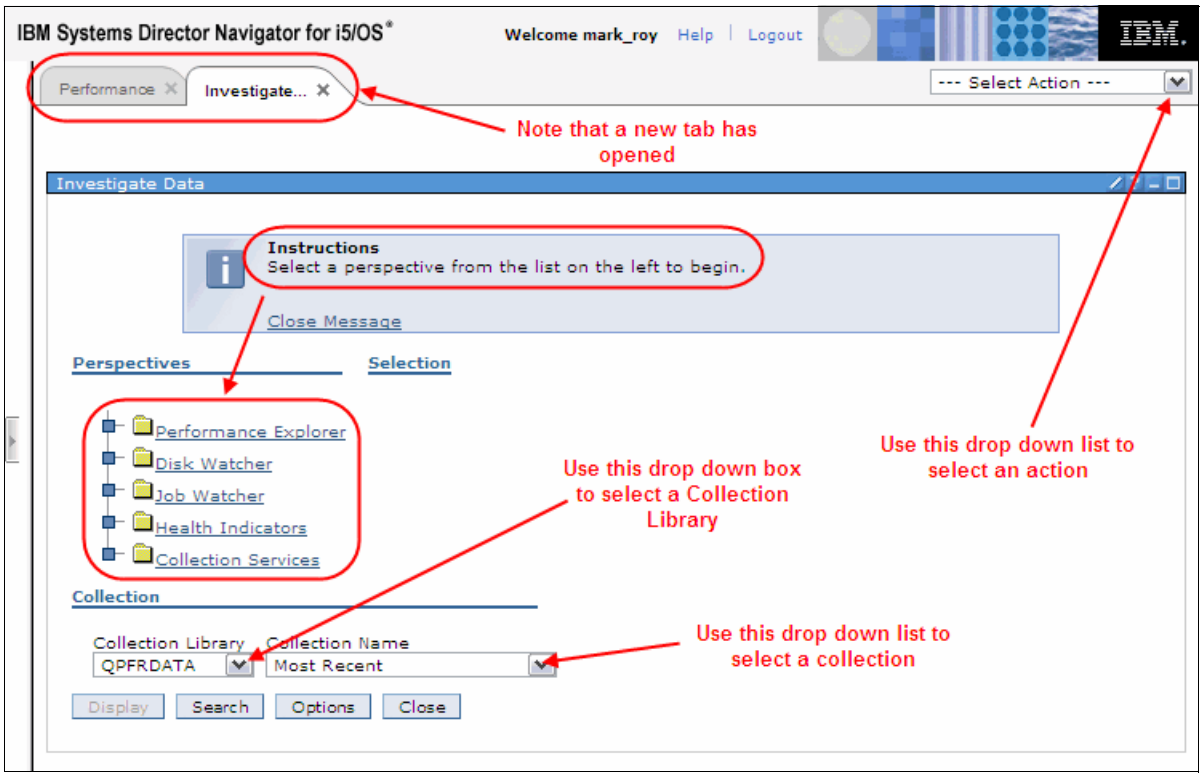


Figure 5-5 Investigate Data window

Looking at Figure 5-5, starting at the top left and going counter-clockwise, you:

- Will notice that the Investigate Data pane is opened in a new tab.
- See a message box, which contains instructions. You can close this message box by:
  - Selecting **Close Message**.
  - Selecting the **Options** button in lower part of the pane and clearing the **Show Help** check box on the subsequent pane (not shown). Use the same option to re-enable the display of the message box.
- Can select a *perspective* from the *perspectives tree*.
- Can use a drop-down menu to select a Collection Library. The drop-down menu contains all the libraries that you are authorized to access that contain any type of collection.
- Can use a drop-down menu to select a Collection Name:
  - If you have not yet selected a Perspective, the list contains all collections of any type in a library.
  - Once you select a Perspective, only collections that are valid for the selected perspective are listed.

- Can select an action:

Close Page	Closes the current page.
Add to My Startup Pages	My Startup Pages displays all pages that will be automatically launched when logging into IBM Systems Director Navigator for i.
Manage Open Pages	Manage Open Pages displays all open pages, where you can close individual, or all, open pages.

This section has introduced the terms *perspective* and *perspectives tree*. We will now explain some of the new terminology.

First, expand the Health Indicators content package by clicking the expand or collapse icon to the left of the Health Indicators link, as shown in Figure 5-6.

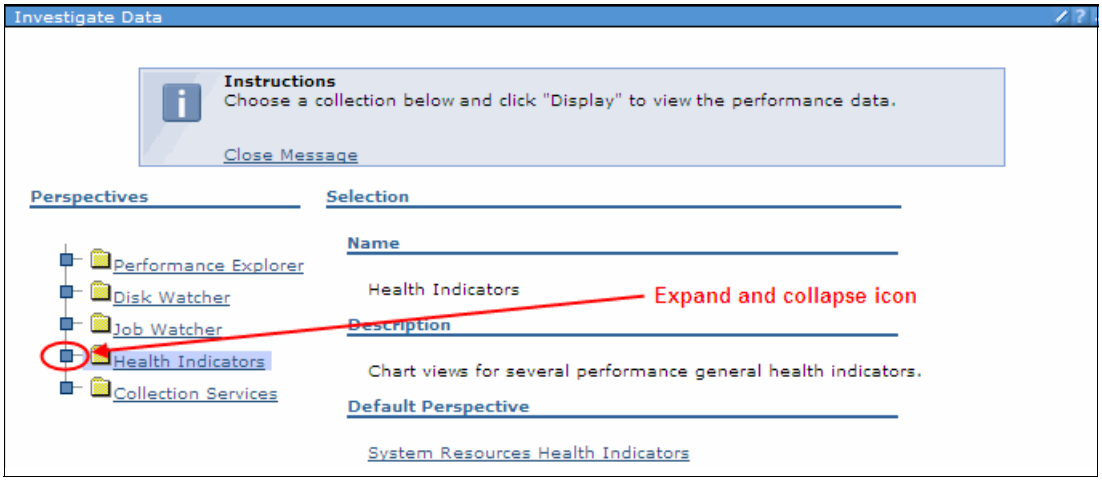


Figure 5-6 Expand and collapse icon

You will be presented with an expanded Investigate Data pane, as shown in Figure 5-7.

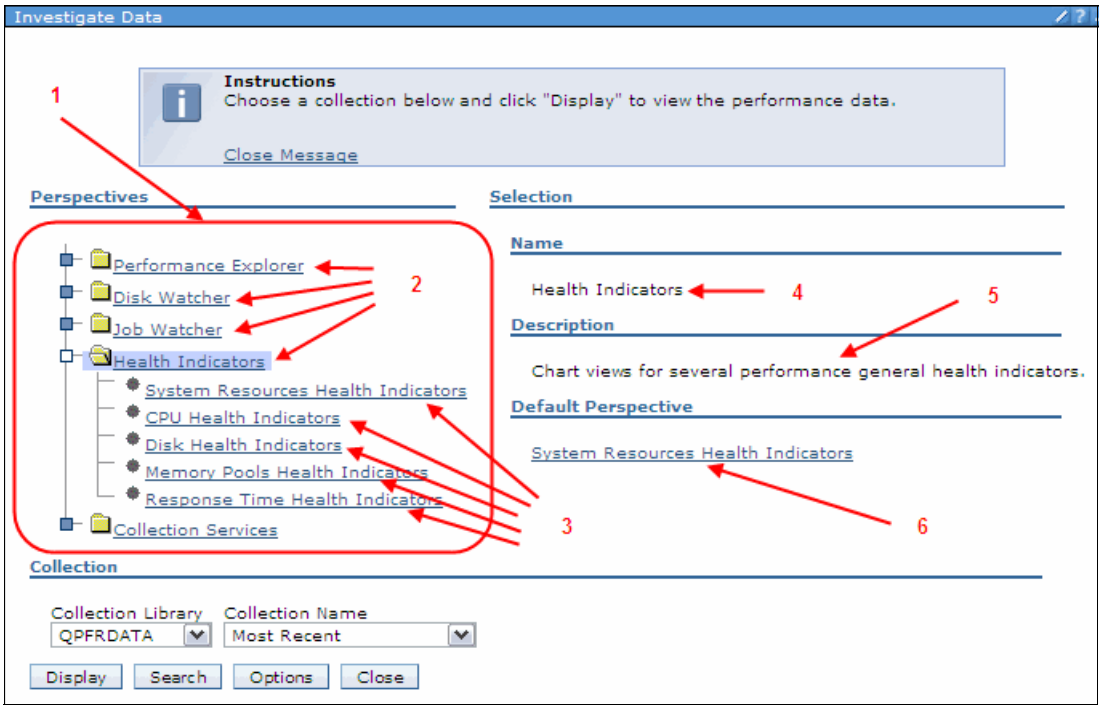


Figure 5-7 Investigate Data page elements

In Figure 5-7 we highlight:

1. The *Perspectives tree*, which displays a hierarchical list of all the perspectives that are available.
2. *Content packages*, which are used to group similar perspectives together.
3. *Perspectives*, which are logical groupings of similar or related tables or graphs that benefit from being displayed side-by-side for reference or context.

In addition:

1. When a content package or perspective is selected, this shows its name.
2. When a content package or perspective is selected, this shows its description.
3. When a content package is selected, this shows the name of default perspective for the content package.

We will now open a graph in a perspective and discuss the navigation tools in a graph pane.

Collapse the Health Indicators sub-tree by clicking the expand or collapse icon to the left of the Heath Indicators content package. This step is not necessary; it simply helps to unclutter the pane.

Next, expand the Collection Services content package. This will reveal the Collection Services perspectives, as shown in Figure 5-8. Then select the **CPU Utilization and Waits Overview** perspective and click the **Display** button, as shown in Figure 5-8.

Note: When you select the CPU Utilization and Waits Overview perspective, the Web page will refresh and update the Selection information on the right side of the pane.

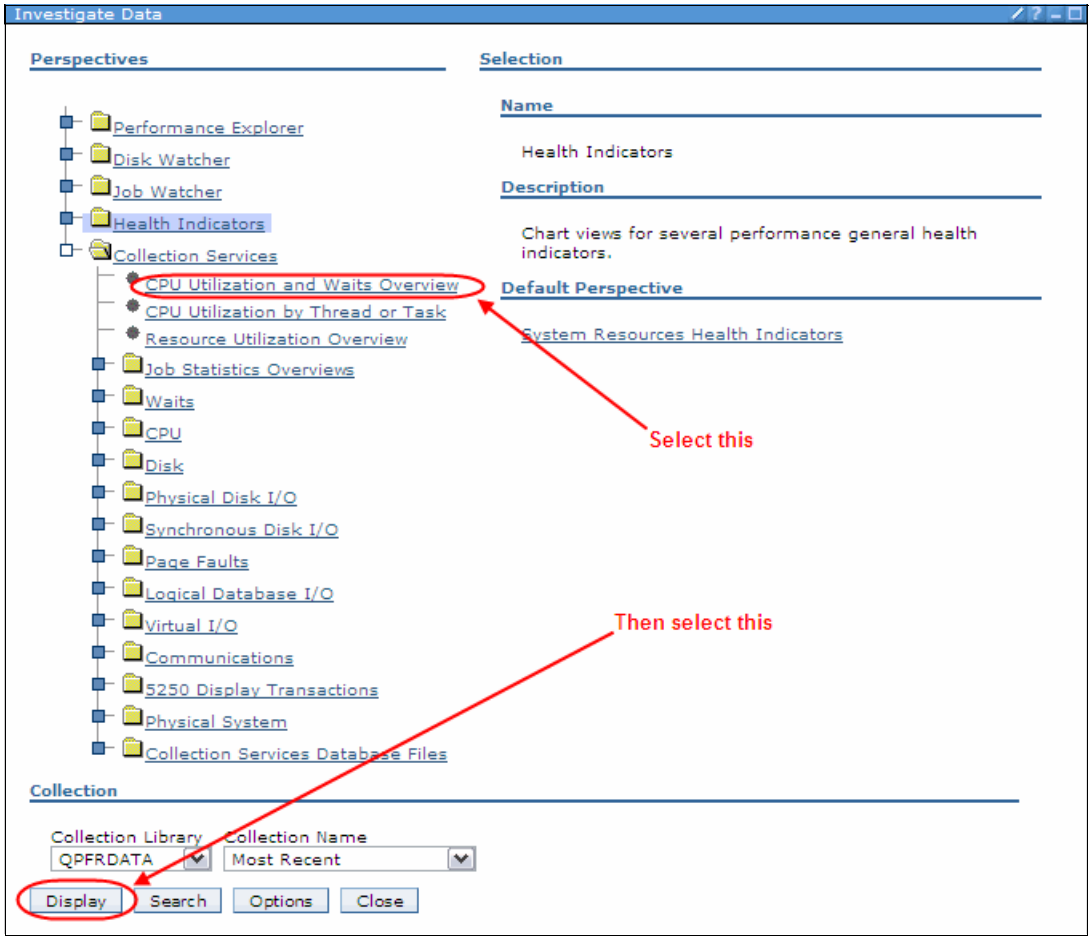


Figure 5-8 Collection Services perspectives

The CPU Utilization and Waits Overview perspective opens, which, by default, is displayed in graph mode. The perspective is shown in Figure 5-9.

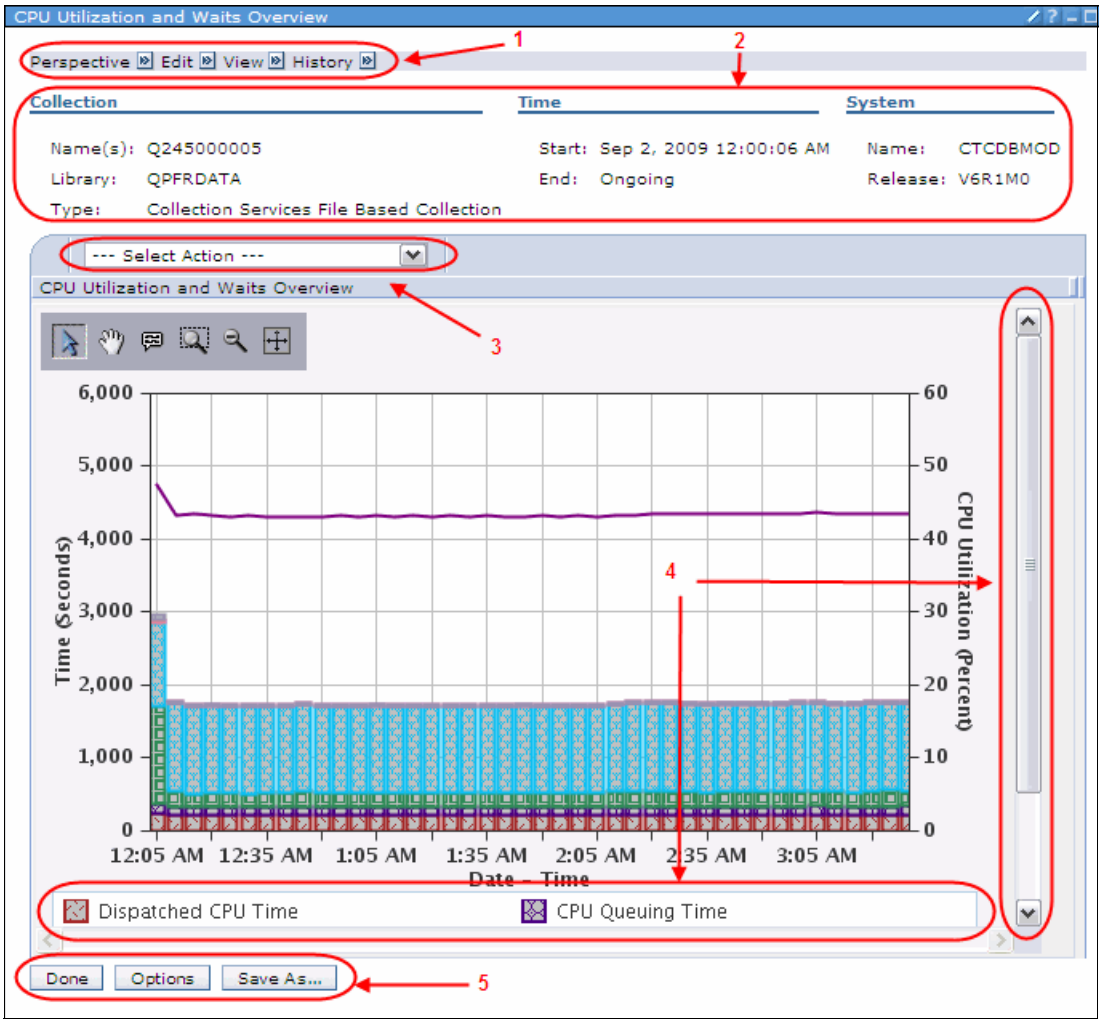


Figure 5-9 CPU Utilization and Waits Overview perspective element

In Figure 5-9, we highlight the following elements:

1. These icons display the menus for:

   Perspective You can save this perspective as a custom perspective or close the perspective.

   Edit Change options for the perspective, such as graph or table mode, set table sizes, and more.

   View Hide or display the *context*. The context is the information about the collection being investigated.

   History Display a list of perspectives you have recently displayed.

2. The context for the perspective.
3. A drop-down menu of other related perspectives that can be selected.
4. Note that the graph legend in Figure 5-9 only shows two color/pattern labels in the lower portion of the pane. You can use the scroll bar on the right side of the pane to display more.

5. These three buttons are shortcuts to some of the more frequently used menu functions.

The CPU Utilization and Waits Overview perspective also contains the graph manipulation toolbar, as shown in Figure 5-10.

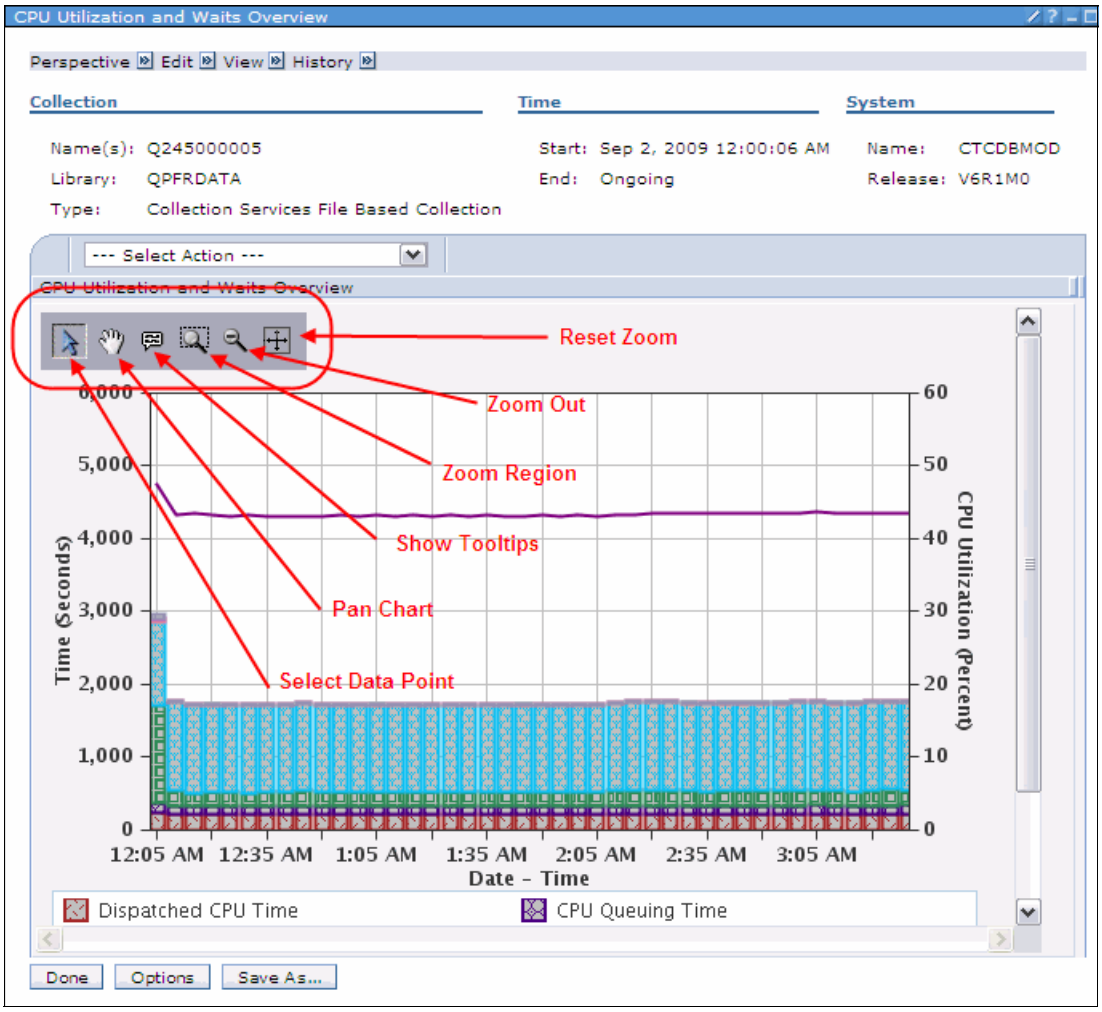


Figure 5-10 Graph manipulation toolbar

This toolbar has the following elements:

Reset Zoom Selecting this icon causes the graph to zoom out. You notice in Figure 5-10 that the time line for the graph starts at around 12:05 a.m. and ends around 3:45 a.m. However, the collection contains data after 3:45 a.m. If you select the Reset Zoom icon, the graph will be redisplayed and it will contain the complete timeline of the collection.

Zoom Out Each time the Zoom Out icon is selected, the graph will be redisplayed with an increasing number of time intervals. Repeated selection will eventually zoom out to show the complete timeline of the collection.

Zoom Region The Zoom Region icon is used to zoom in on data in the graph. Selecting the Zoom Region changes the mouse cursor to this shape:

You can then click and hold down the left mouse button, then drag the mouse over the data you wish to zoom in on. You can zoom horizontally (to view less time intervals by widening the bars) or vertically (to reduce the scale and increase the height of the bars).

Show Tool Tips This icon enables the Tool Tips function. With Tool Tips enabled, when you hover the mouse cursor over different colored sections in a bar in the graph, a small window will open. It shows interesting metrics about that point in the chart.

Pan Chart The Pan Chart icon allows you to slide the time intervals left or right. Selecting the Zoom Region changes the mouse cursor to this shape:

You can then place the mouse cursor anywhere within the graph; click the left mouse button and hold it down, then drag the mouse cursor left or right. When you release the mouse key, the time intervals will be adjusted accordingly.

Select Data Point Selecting a data point allows you to show the value of the data in the different colored sections of a bar. To enable it, click the Select Data Point icon and then click the bar section that you wish to reveal. The number that is displayed is the length of time for the selected section of the bar. Points can also be selected on line series.

In the most cases, once an icon is selected, the corresponding function can be repeated until another icon is selected.

We will provide examples of using the graph manipulation icons in the 5.3, “Investigating data” on page 91 section.

5.3 Investigating data

In this section, we discuss how to use the perspectives to investigate performance data. We use more than one tool. As stated in 3.3.2, “Tool usage” on page 39, there is often no right or wrong tool to use on a given set of data at a given point in time during an investigation. In our example, we provide guidance about why a particular tool was used.

For our investigative purposes, we generated a workload and collected performance data. We also created performance problems that can occur in production environments, so we can use the tools in way that will have some meaning in real environments.

> **Note:** When collecting our performance data, we used three collectors at the same time: Collection Services, Job Watcher, and Disk Watcher. We did this because we needed the different collections for this example. When investigating a real problem, it is more likely you would use tools serially. This will become apparent as we work through the example.

Our investigation was initiated after some fictitious users of the system reported intermittent poor response times. Management has requested that we identify the causes and provide recommendations about how to address these issues.

5.3.1 Investigate using Health Indicators and Collection Services

In general, the first thing we look at is the overall health of the system. We perform this action to see if a system is heavily constrained, that is, if the workload on the system is simply too much for the installed configuration. In this scenario, erratic response times are common and can often be resolved by upgrading the hardware. To check this situation, we use the Health Indicators perspective.

Before we can use the perspective, we need to decide which collection to use. If a system is constrained most of the time, it is likely that any type of collection will suffice. However, in our example, the users who reported the problem also mentioned the approximate date and time the poor response times were experienced. Using that information, we decide to use the collection that was running at that time. For information about how to find the name of a collection for a given period, refer to 4.6.2, “Working with collections” on page 74.

In our example, we use Q244000002, which is in the default collection library QPFRDATA. Now that we know the collection name, we look at the Health Indicators perspective for that collection. To do this, we perform these steps:

1. Expand the Health Indicators content package.
2. Select **System Resources Health Indicators**.
3. Use the Collection Name drop-down menu to select the collection we need.
4. Select the **Display** button.

These steps are shown in Figure 5-11.

Note: In the remainder of this chapter, we have suppressed the message box and context display in many of the screen captures.

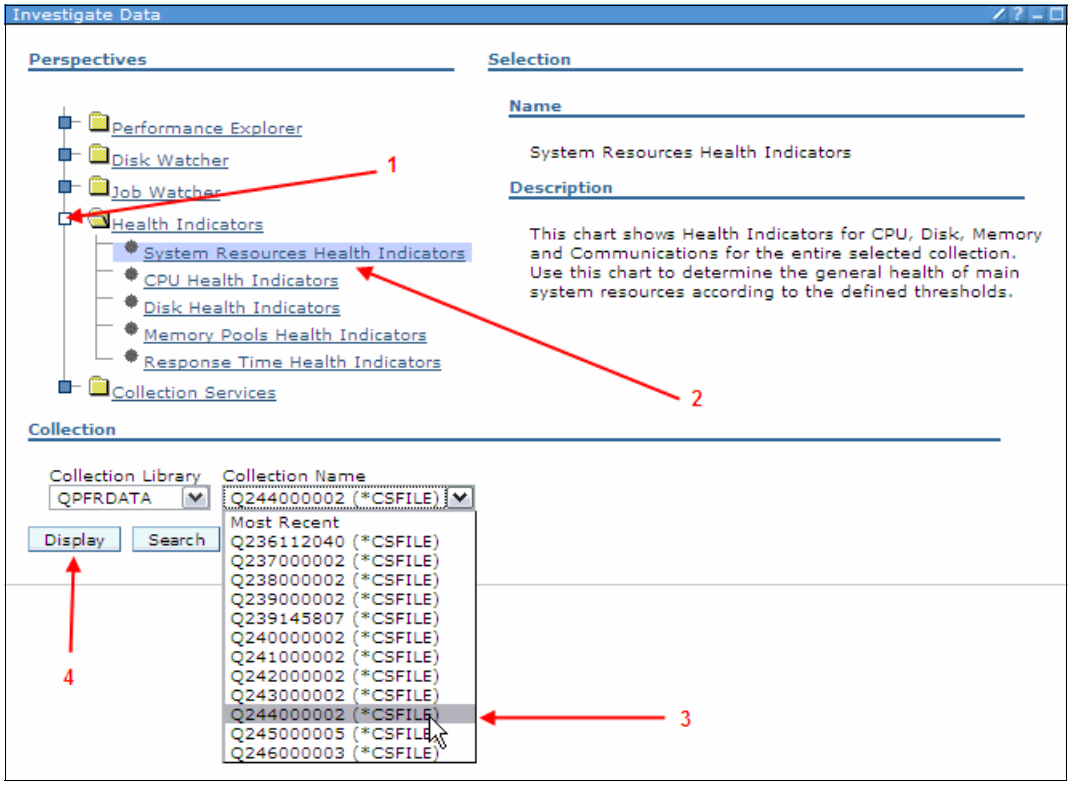


Figure 5-11 System Resources Health Indicators

Note: The Collection Library and Collection Name lists are built when you start working with the Investigate Data tab. If additional collections are created while you are using the Investigate Data tab, the new collections will not appear in the drop-down menus. To rebuild the lists so that they will include the new collections, you must exit the Investigate Data tab and re-enter it.

The System Resources Health Indicators perspective is displayed, as shown in Figure 5-12.

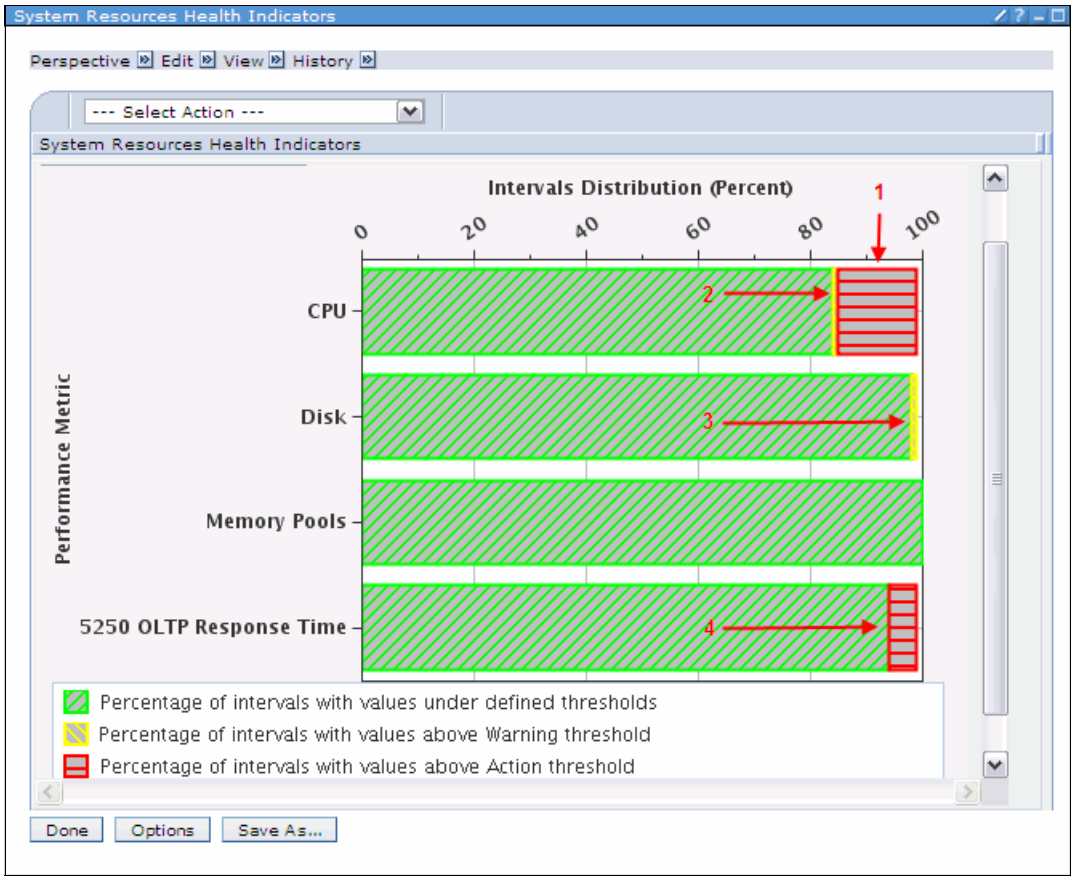


Figure 5-12 System Resources Health Indicators

We glean some helpful information from this graph:

1. The CPU is not constrained most of the time. It it were, the horizontal bar would show more red than green. However, there was a period (or periods) where the CPU was above the Action threshold.
2. For a relatively short period or periods, the CPU was above the Warning threshold but below the Action threshold. This is evident from the thin yellow section on the bar. This implies that the time between the CPU going from below thresholds to above the Action threshold is relatively short. In other words, whatever is consuming the CPU excessively consumes and releases it quickly. This knowledge may come in useful as we travel down the investigation path.
3. The disk health indicators show that they were over the Warning threshold occasionally but did not exceed the Action threshold.
4. it is evident that, on occasion, 5250 response times are excessive. We were not advised whether the poor response times were only in 5250 applications or Web applications, or both. We investigate and discover that the 5250 response times correlate with what we are seeing in the 5250 OLTP Response Time.

The Memory Pools bar in the chart is all green, so memory constraints do not appear to be an issue and is unlikely to be a contributing factor to our intermittent poor response times.

Note: If you wanted to change the thresholds for the Health Indicators, you can use the drop-down menu near the top left and select **Define Health Indicators**.

Next, we will drill down to CPU Health Indicators, as shown in Figure 5-13.

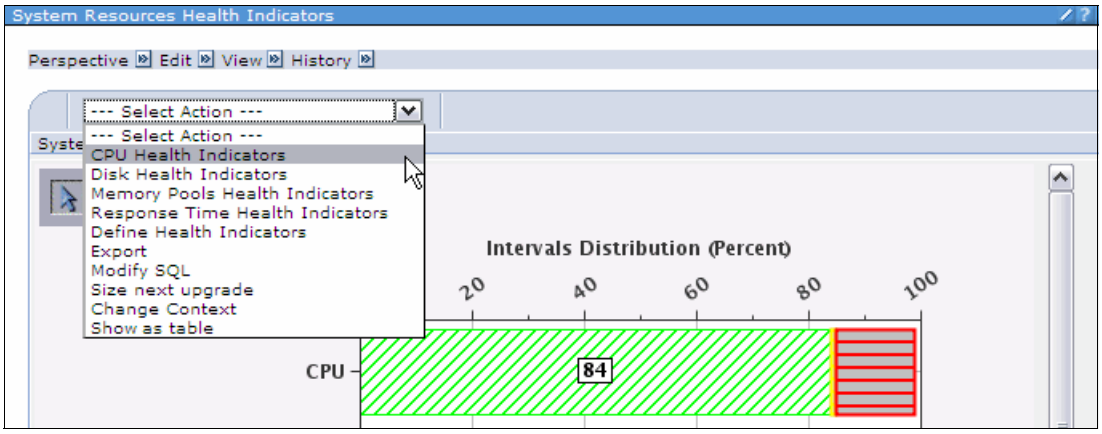


Figure 5-13 Selecting CPU Health Indicators

We are presented with the CPU Health Indicators perspective, as shown in Figure 5-14.

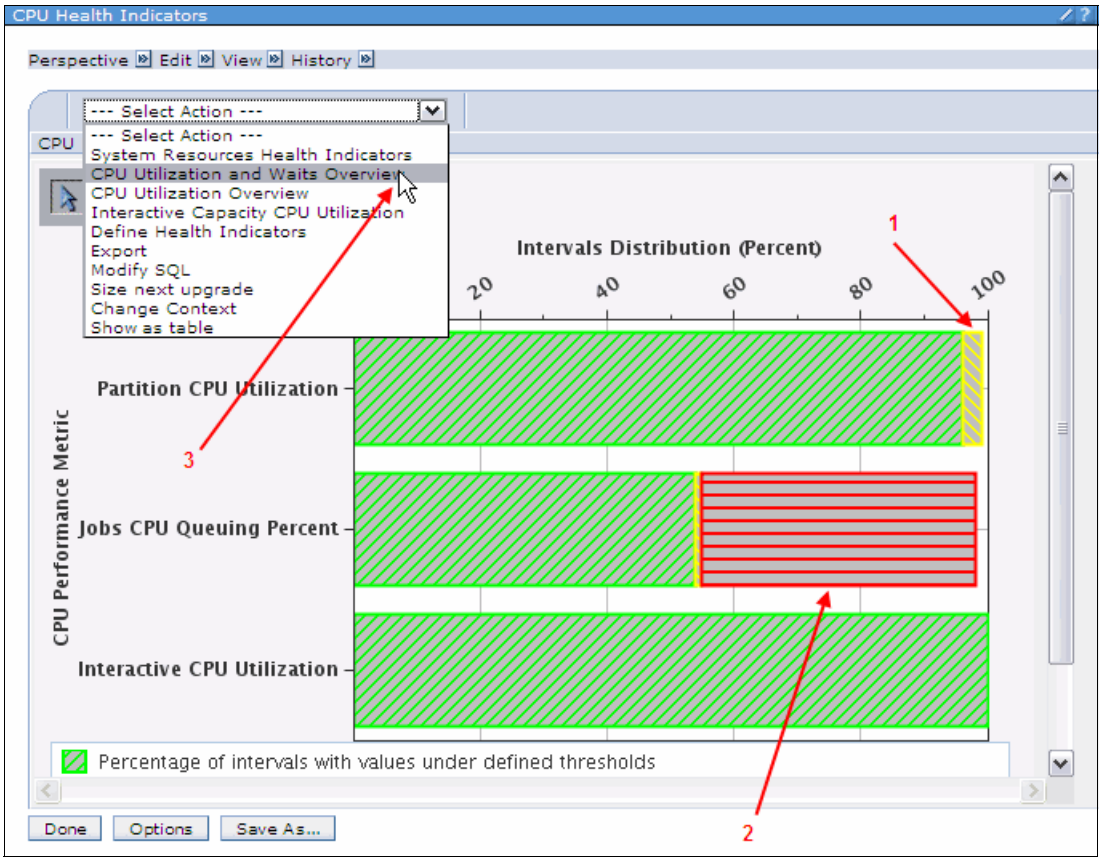


Figure 5-14 CPU Health Indicators

In the CPU Health Indicators perspective, we can see:

1. The partition CPU utilization was over our Warning threshold for a relatively short period of time, but did not exceed the Action threshold.
2. Almost 50% of the time, jobs were queuing for the CPU.

   It appears that:

   – One or more jobs were hogging the CPU, causing the excessive queuing.

   – The Interactive CPU Utilization did not exceed thresholds, so it is unlikely that the reported poor response time transactions were slow because of excessive CPU utilization within the transactions.

   – Given that almost half of the collection time was spent in CPU queuing, it is unlikely that the queuing is a direct cause of our reported intermittent poor response times. If it was, we would expect significantly more reports of poor response times.

   So, what do we know now? It is unlikely that constraints on the system resources are the cause of our poor response times. Next, we will start to look at the waits on the system to see if any of those will help us track down the root cause.
3. We now select the CPU Utilization and Waits Overview perspective from the drop-down menu, as shown in Figure 5-14 on page 95.

We are presented with the CPU Utilization and Waits Overview perspective, as shown in Figure 5-15.

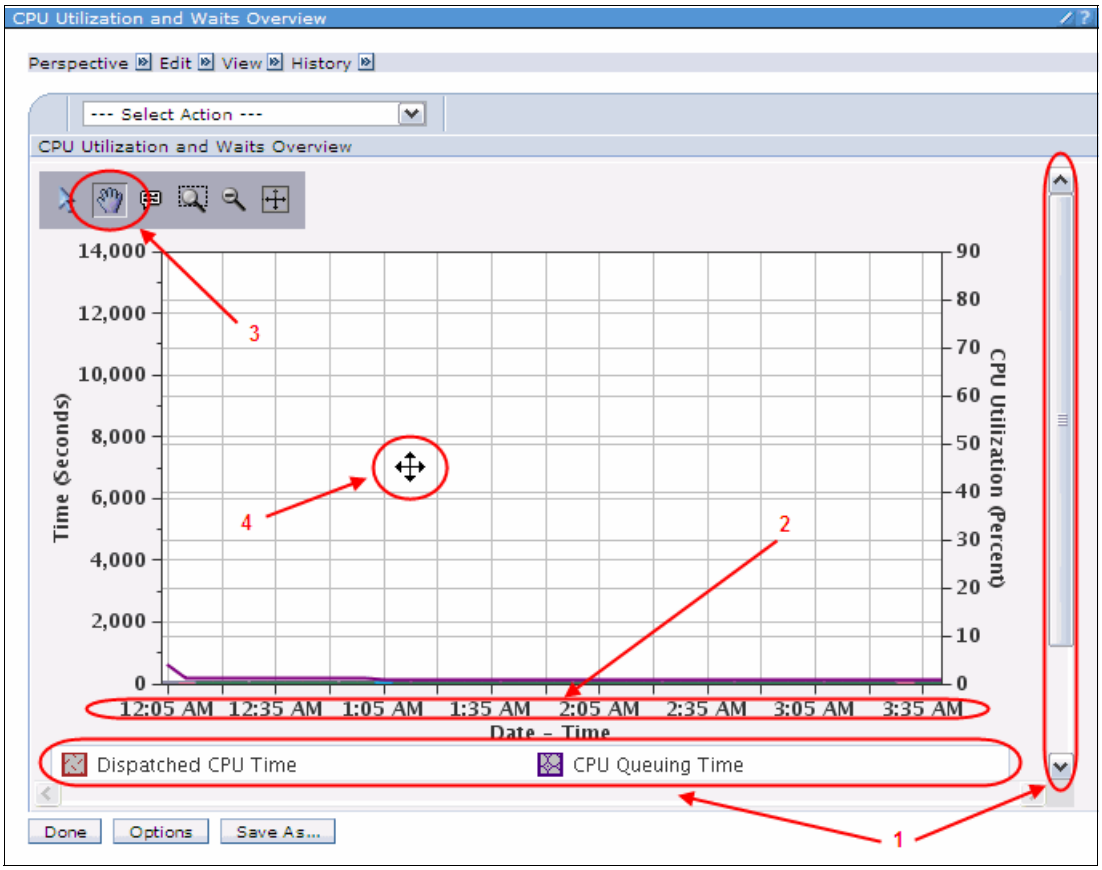


Figure 5-15 CPU Utilization and Waits Overview

Note: We have now drilled down to the Collection Services perspectives. We could have travelled here by two paths. One could have been through the Collection Services content package on the perspectives tree, or, through the way we have shown in this example. The point here is that there are multiple ways to get to most perspectives.

In Figure 5-15 on page 96, we need to make some adjustments to the display content, such as:

1. You notice that on the legend across the bottom of the graph, only two labels are displayed. It is likely that there are more available. To confirm this, look at the scroll bar on the right. If it is scrollable, use it to display additional labels.
2. When opening this perspective, the default is to show data from the beginning of the collection. In our example, the collection started around 12:05 a.m. Again, by default, the graph only shows a fixed number of collections, not the entire collection. This is where we can use the pan function from the graph manipulation icons.

   In our example, the users reported that the poor responses occurred somewhere between 2:30 p.m. and 4:00 p.m. If we did not know that fact, we can still use the pan function to step through the collection and, while doing so, keep looking for any interesting spikes or patterns. As you can see in our collection, not much was happening on the partition early in the morning. Even though we know the approximate time we want to look at, we still use the pan function to arrive there.

   Note: An alternative could have been to use the Create Performance Data (CRTPFRDTA) command to create a new collection that only shows the time period in which we are interested. We could then use that new collection and reduce the amount of panning we need to do. For more information about the CRTPFRDTA command, refer to Figure 4-13 on page 70.

3. To start the pan function, we first select the Pan icon.

4. You notice that the mouse cursor has changed, so that we know we are now in panning mode. We then place the mouse cursor near the right side of the graph, click and hold down the left mouse key, and then drag the mouse cursor to the left. You will notice a black outline box appearing in the graph frame. This is used to provide a reference point so you can determine how far you want to pan. The thinner the box is, the more data is panned. This is shown in Figure 5-16.

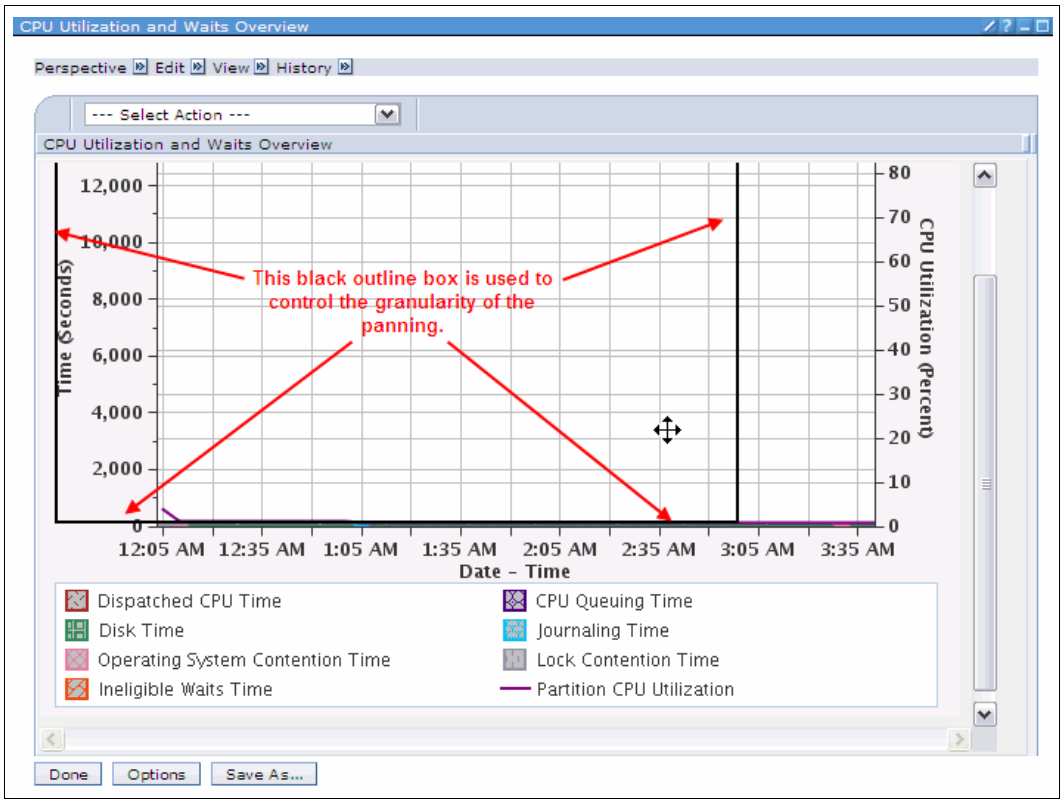


Figure 5-16 Panning

You also notice in Figure 5-16 that we have used the scroll bar to expose more labels in the graph legend.

We now start to pan to approximately 2:30 p.m. After a few panning operations, we get to the period in the collection that we are interested in, as shown in Figure 5-17.

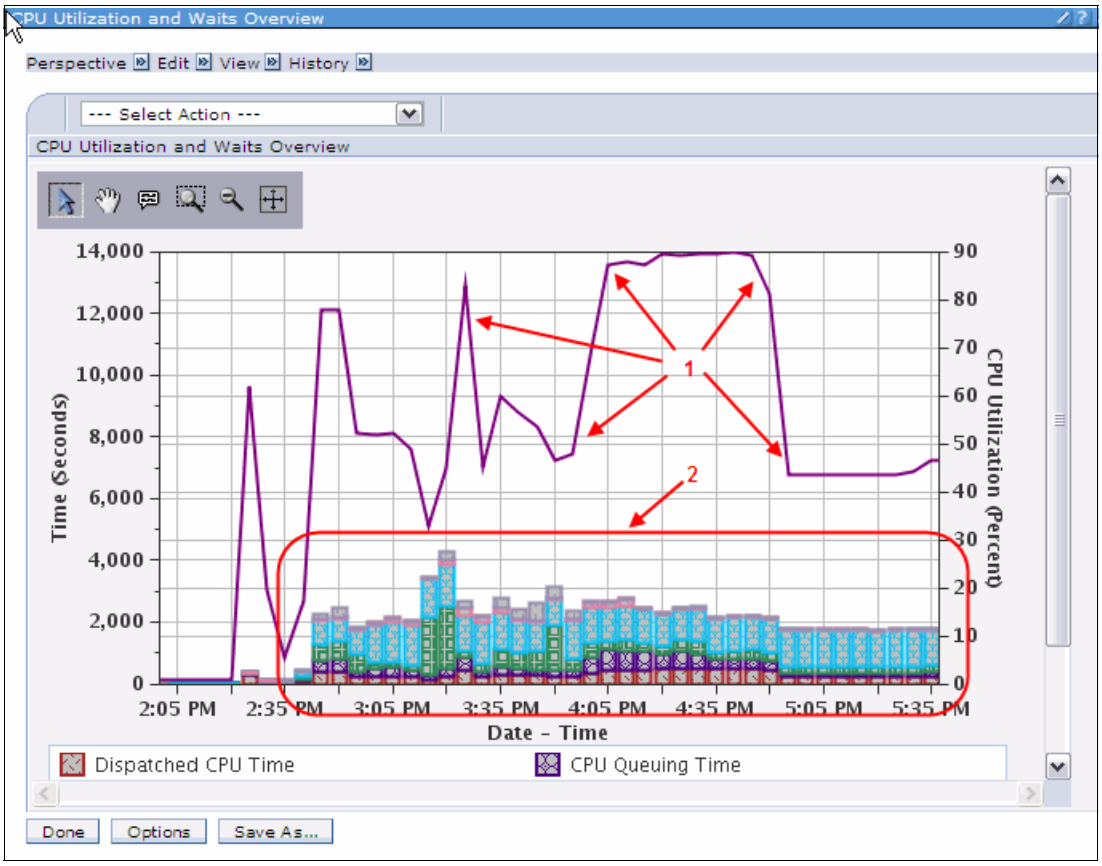


Figure 5-17 CPU Utilization and Waits Overview panned

The CPU Utilization and Waits Overview perspective reveals two interesting events:

1. The CPU utilization of the partition was peaking quite high for both short and also sustained periods.
2. There are a few wait buckets in use, which is evident by the coloration of the bars.

So now we need to work out whether our intermittent poor response times are CPU consumption related or wait related. We already suspect that they are not CPU consumption related, so we will now zoom in on the waits to see what types and duration they were. The process of zooming in is shown in Figure 5-18.

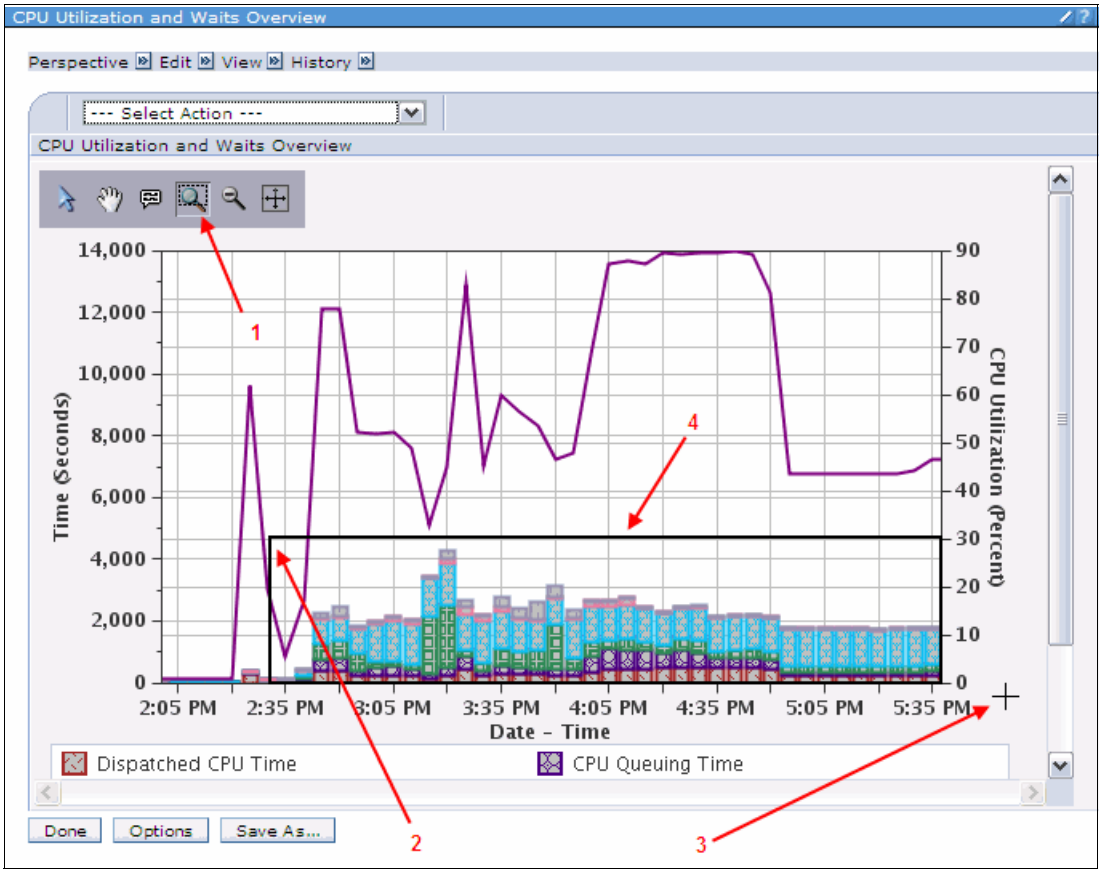


Figure 5-18 Zooming in

To zoom in, we take the following actions:

1. To start the zoom function, select the Zoom Region icon.
2. We move the mouse cursor to the top left of the area we want to zoom in on.
3. We then click and hold down the left mouse key, then drag the mouse cursor to outline the area we wish to zoom in on.
4. You notice a black outline box appearing in the graph frame. This is used to provide a reference that you can use to see what you are about to zoom in on.

   You also see that the mouse cursor has changed. This happened when the Pan icon was selected. This new cursor indicates we are in Zoom Region mode.

The results are shown in Figure 5-19.

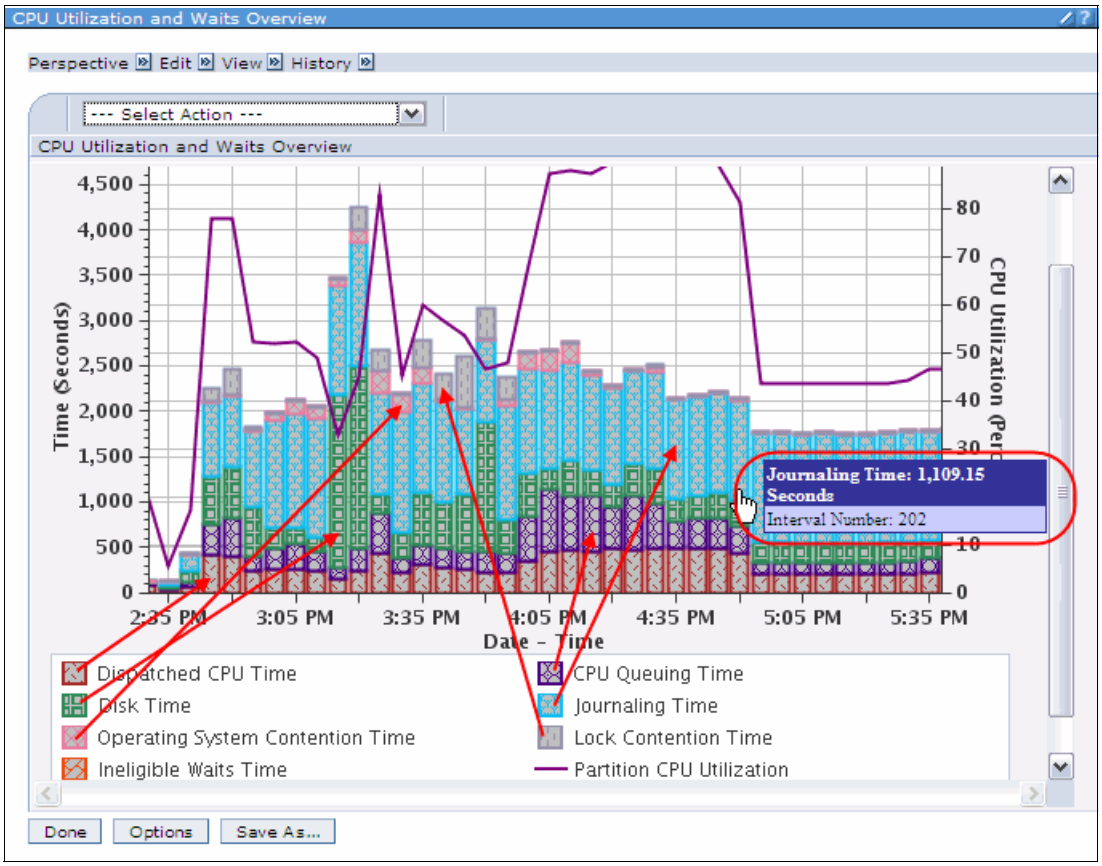


Figure 5-19 CPU Utilization and Waits Overview zoomed in

Now that we have zoomed in, we can see the breakdown of the waits much more clearly. You can discover which wait type each color in the bar represents by either looking at the legend or by selecting the Show Tool Tips icon and hovering the mouse cursor over the color in the bar. We show both methods in Figure 5-19.

Looking at the pattern of the graph, the first thing that catches our attention is the light blue sections, Journaling Time. You can see that this is consistently a significant component of the wait time in each interval. Could this be the cause of our intermittent poor response times? Once again, probably not. Given the consistency of the lengthy Journaling waits, if this were the root cause, we would expect constantly poor response times.

Even if Journaling waits are not part of our reported problem, it indicates that there may be an unrelated problem of which we are not aware. In most of the intervals, Journaling waits are 50% or more of the wait time. If the jobs that are experiencing these waits spend a lot of time in disk I/O, there may be opportunities to significantly reduce their run time. This is a prime example of using the collection analysis tools to discover unknown problems.

We set aside the Journaling waits for now and continue with our investigation into intermittent poor response.

We notice that in two intervals around 3:20 p.m., the Disk wait time is significantly higher than in other intervals. Could this be our problem? Probably not. The users who reported the poor response times indicated they were intermittently occurring over the whole 2:30 p.m. to 5:.00 p.m. time period. The spike in Disk wait time may have been a factor if there was a poor response transaction around 3:20 p.m., but probably not a factor at other times.

We keep looking. The next thing we notice is the Lock Contention wait time. It is quite noticeable between 3:20 p.m. and 3:50 p.m. If we look closely, we can also see it was occurring, to a lesser extent, after that time. We use the Select Data Point icon to view the total time that was spent in Lock Contention in one of the intervals. To do this, we click the icon and then click the data point we wish to reveal, as shown in Figure 5-20.

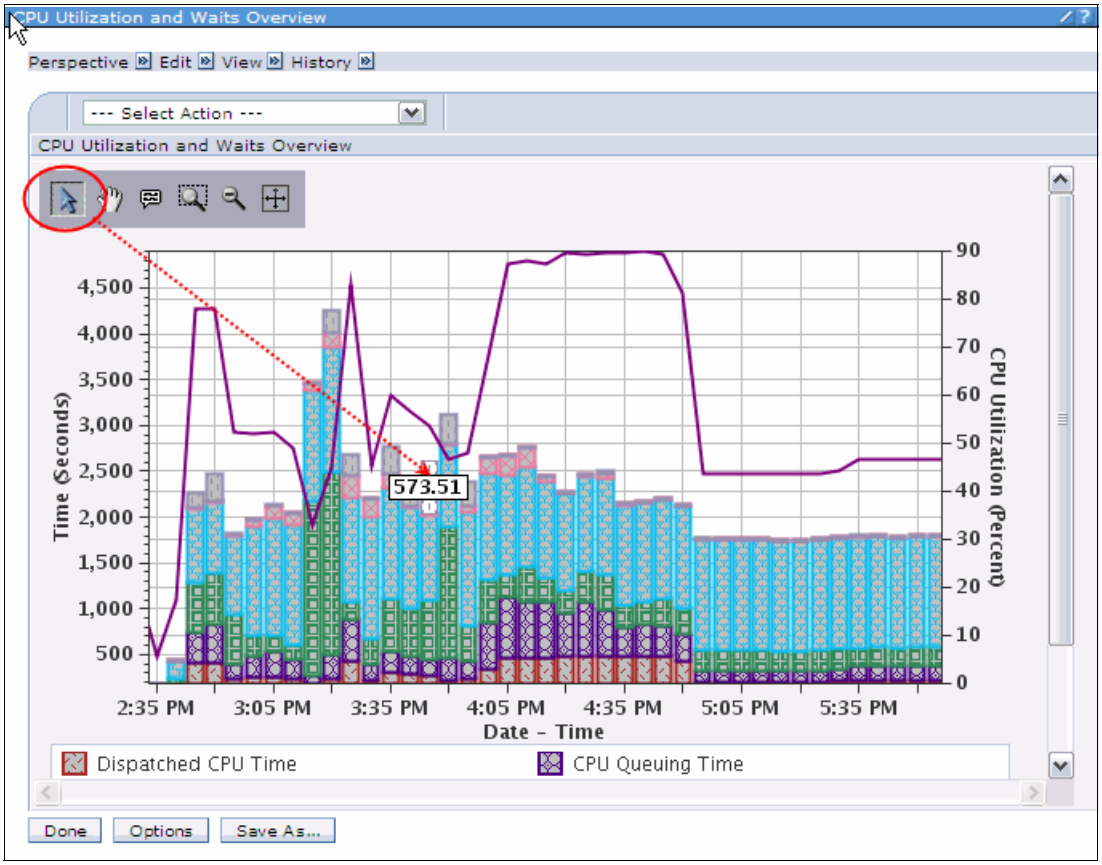


Figure 5-20 Selecting a Data Point

We can see that it is around 573 seconds. Some quick mathematics tells us this is almost one entire interval. In our collection, we collected at 5 minute, or 300 second, intervals. Our next task is to discover what types of locks are occurring and to what extent. To do this task, we select the Seizes and Locks Waits Overview perspective from the drop-down menu, as shown in Figure 5-21.

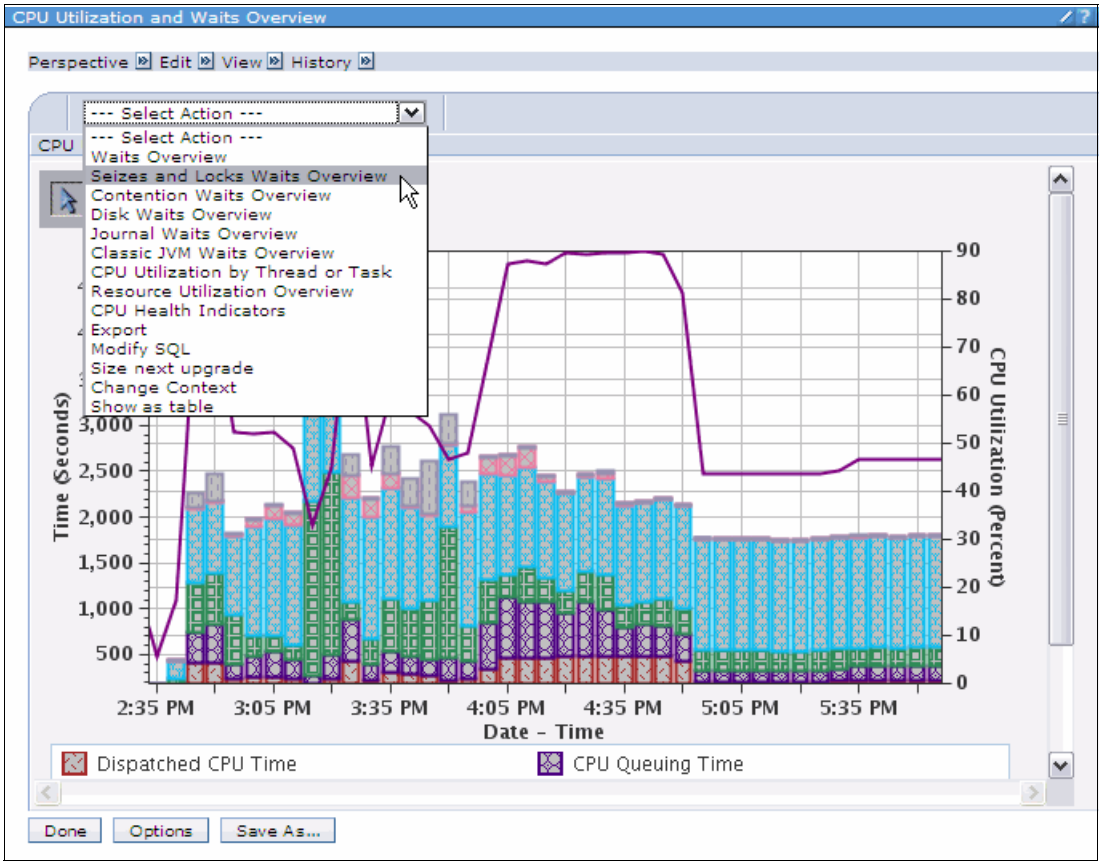


Figure 5-21 Selecting Seizes and Locks Waits Overview

Note: If you select a perspective from the drop-down menu, without first selecting a data point, the resulting graph shows all intervals. If a data point has been selected, the resulting graph only contains the data for the selected interval. At this point in our investigation, we want to graph all intervals, so we must clear the data point. To clear a data point, simply click the data point again (assuming that the Select Data Point mode is still active).

In the resulting graph, we use the zoom function to magnify the bars on the chart. We see the graph shown in Figure 5-22.

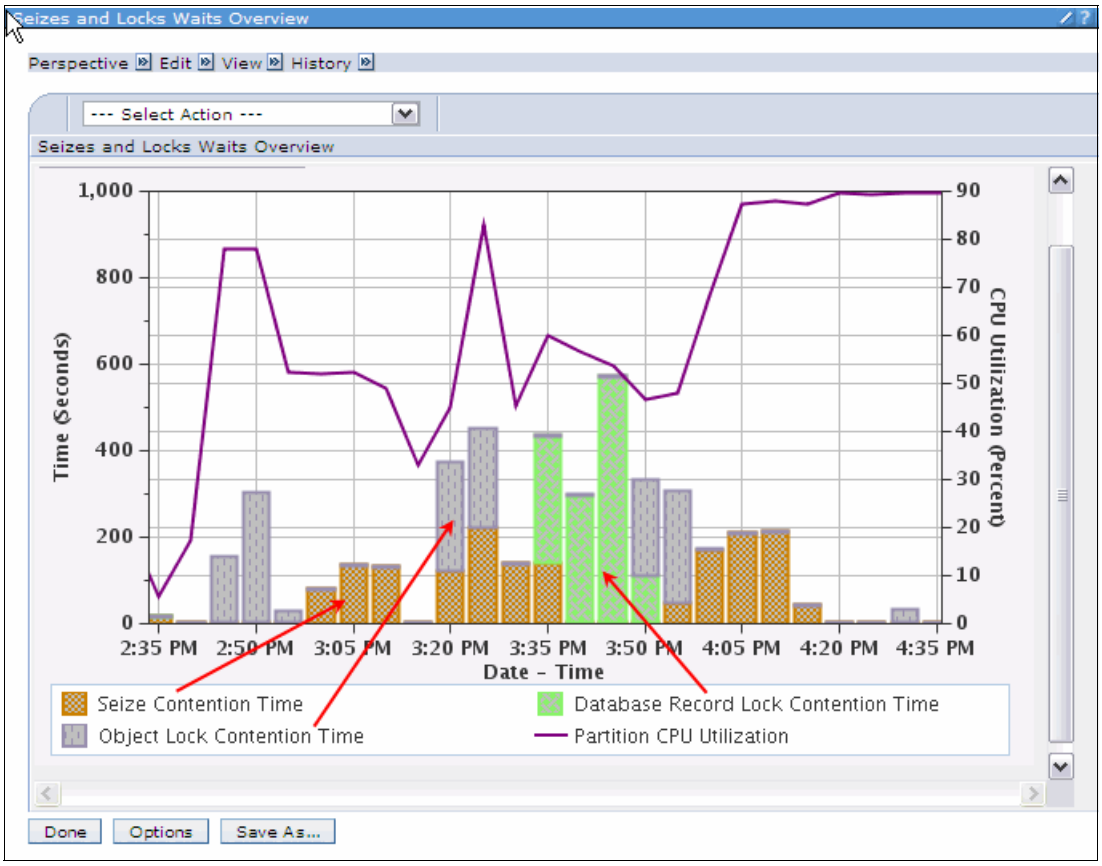


Figure 5-22 Seizes and Locks Waits Overview

You can see that we are experiencing three different types of contention:

Seize A Seize is a lock-wait condition that occurs at the Machine Interface (MI) level. The MI is the interface, or boundary, between the operating system and the Licensed Internal Code.

Object Lock A Object Lock is a lock-wait condition that occurs at the operating system level. It is called an object lock because it appears as a lock on an object, not at the components that comprise a given object.

Database Record A Database Record lock is also a lock-wait condition that occurs at the operating system level. It differs from an Object Lock because only a record in a file is locked, not the entire file object.

Looking at the graph, we can see that all three conditions are having a similar impact in terms of total wait time. In that situation, we would suggest looking at Object Locks and Database Record locks before Seizes. Our reasoning is that application owners are more likely to be able to influence the condition by modifying their application code or environment. Seizes are sometime more problematic to resolve because the application owner has less influence over them. However, if the Seizes and Locks Waits Overview perspective was showing significantly more total wait time in Seizes, that would alter the priorities of the investigation.

We decided to investigate the Database Record locks first. The next thing we want to determine is whether these Database Record locks were occurring in the users' jobs, in particular, those users who initially reported the poor response times. We can do this by selecting the Waits by Job or Task perspective, as shown in Figure 5-23. You need to select the bar for the interval in which we are interested.

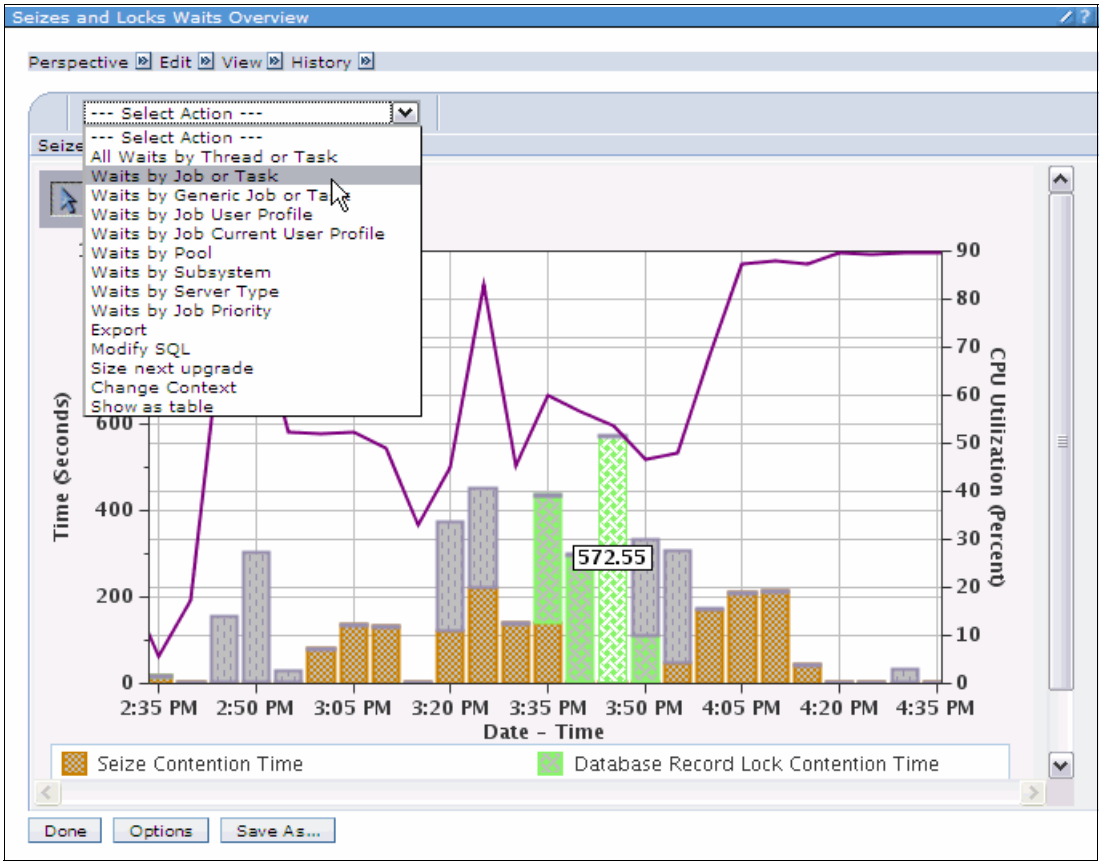


Figure 5-23 Selecting Waits by Job or Task

A different graph format opens, as shown in Figure 5-24.

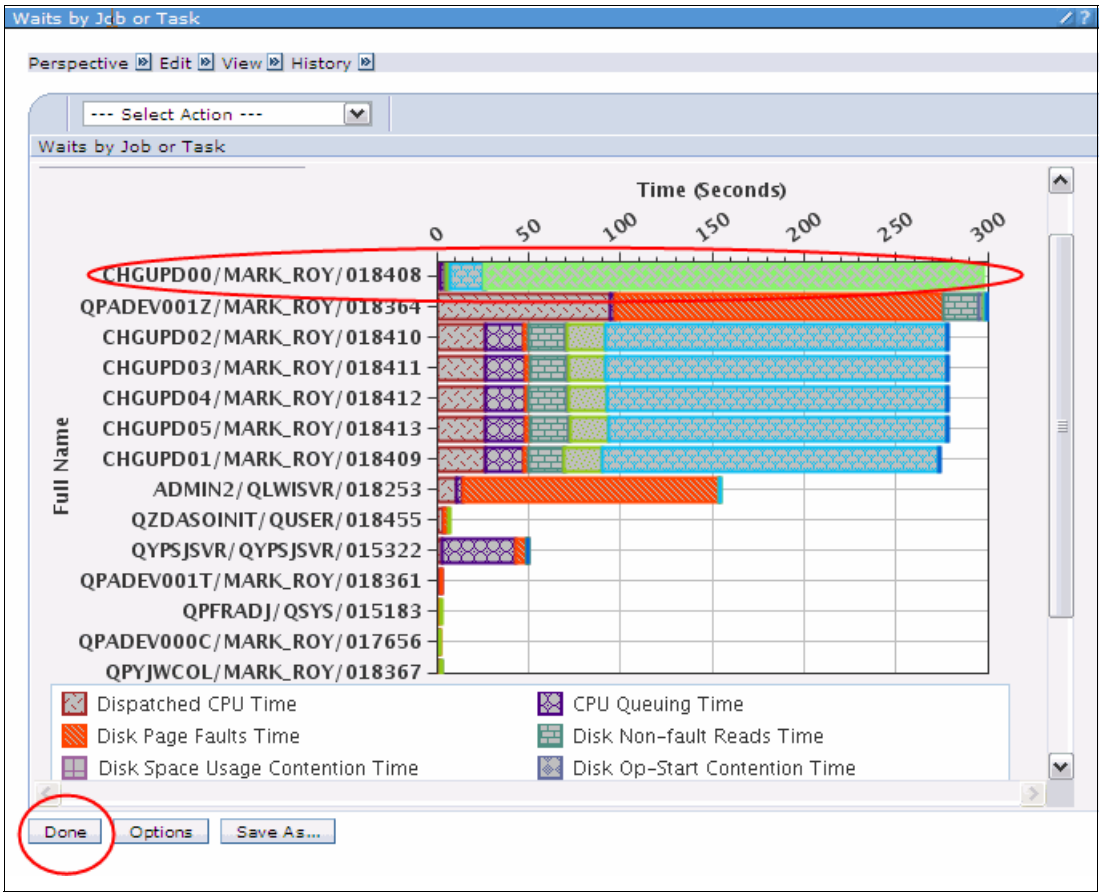


Figure 5-24 Waits by Job or Task: batch job

This chart lists, in descending order, each job's wait time signature. In this example, we see that a job called CHGUPD00 has the most wait time spent in Database Record locks. The user name for the job is MARK_ROY, who is one of the users who reported poor response times. Have we found our root cause? No. We happen to know that in this partition, Telnet devices use the default QPADEV*nnnn* naming convention, so this job is probably a batch job. We try again. We select the **Done** button to return to the Seizes and Locks Waits Overview perspective.

Rather than displaying the job data for each interval individually, we decide to select a number of intervals exhibiting Database Record lock waits. We do this in a similar way to before, that is, by clicking the Select Data Point icon and then clicking the data points we wish to reveal. If an unwanted Data Point is still showing, you can click it to clear it. We then select the **Waits by Job or Task** perspective from the drop-down menu.

The graph shown in Figure 5-25 opens.

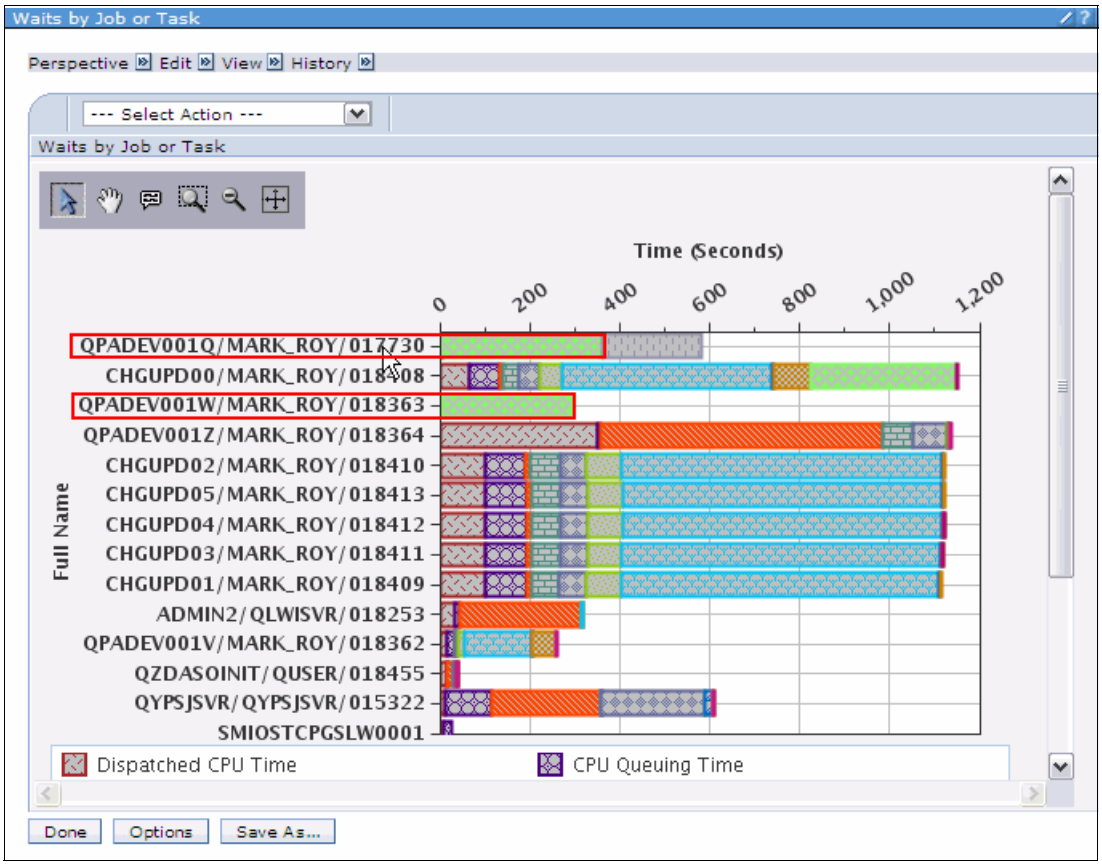


Figure 5-25 Waits by Job or Task: interactive job

In this chart, we see:

- Two jobs for one of the users who reported the poor response times.
- Both jobs are interactive.
- Both jobs exhibit lengthy Database Record lock waits.

We appear to have found one of the causes of the reported poor response times. So, what do we need to discover next? Knowing that we have a Database Record locking conflict issues is not enough to resolve them. We need additional information, such as:

- Which program was the user running when the lock occurred?
- What file was the locked record in?
- What job was locking that record, and for how long?
- What program was the locking job running when it held the lock?

The Collection Services data that we have just investigated has served its purpose. It has helped us identify what kind of a problem we have. Now that we know what kind of problem we face, we can select the appropriate tool to investigate that type of problem. In our example, we want to discover more about what a job was doing. The perfect tool for that is Job Watcher.

5.3.2 Using Job Watcher for further investigation

Before we can begin analyzing the root cause of the poor response time using Job Watcher, we have to determine which Job Watcher collection to use. If you have set up Job Watcher to run continuously on your system, you can use the Collections interface to look for the appropriate collection.

Determine the appropriate collection

1. In the Performance tab, click **Collections** to open all the collections that exist on the system.
2. By default, the Collections window displays performance collections created by Collection Services, Job Watcher, and Disk Watcher. Navigating through several hundred collections to find the one we are interested in can be difficult. However, we can narrow down the list by using Filtering function. Click the Show Filter Row icon or select **Show Filter Row** in the Action drop-down menu, as shown in Figure 5-26, and click **Go**.

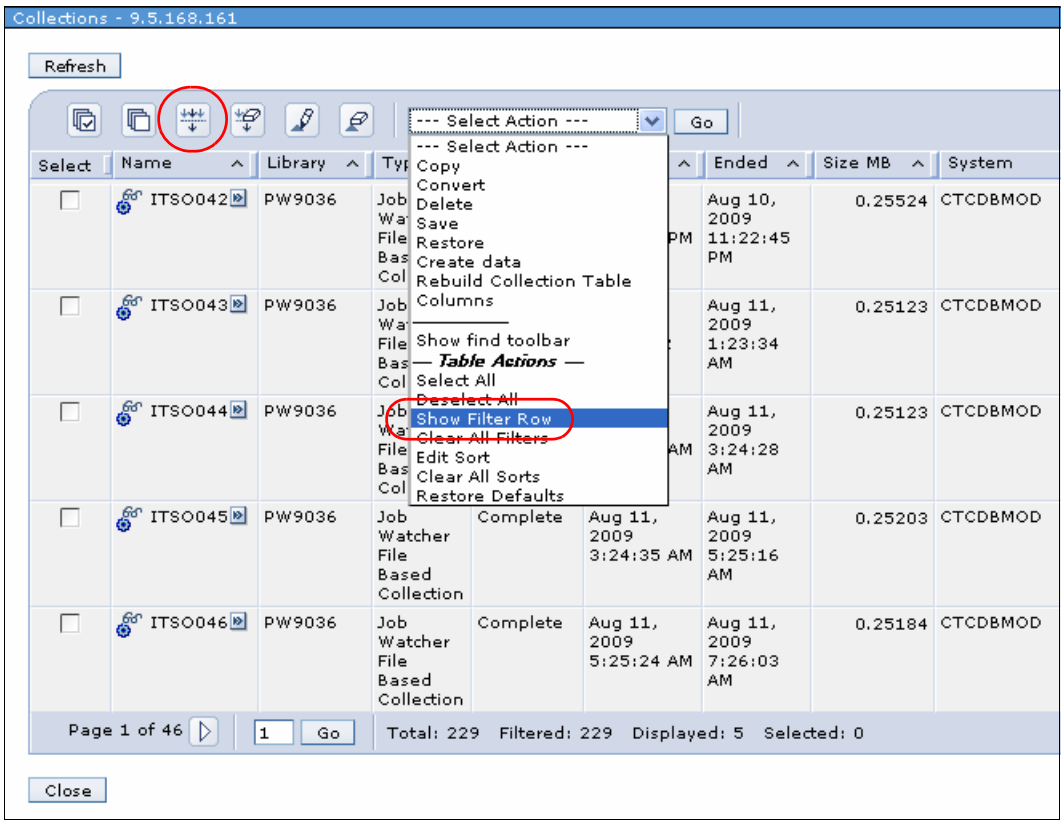


Figure 5-26 Collections window

Note: There are JW/DW/CS Collections tasks under each collector when you look at “Show All Tasks”. These tasks are shortcuts that take you into an already filtered list of collections. If you know you are only interested in one collection type, this is a quick way to get it.

3. As we know the date and time from our earlier analysis, we can filter collections by clicking the **Filter** link in the Started column. In the filter window, select the date range when the problem was reported to have occurred, as shown in Figure 5-27.

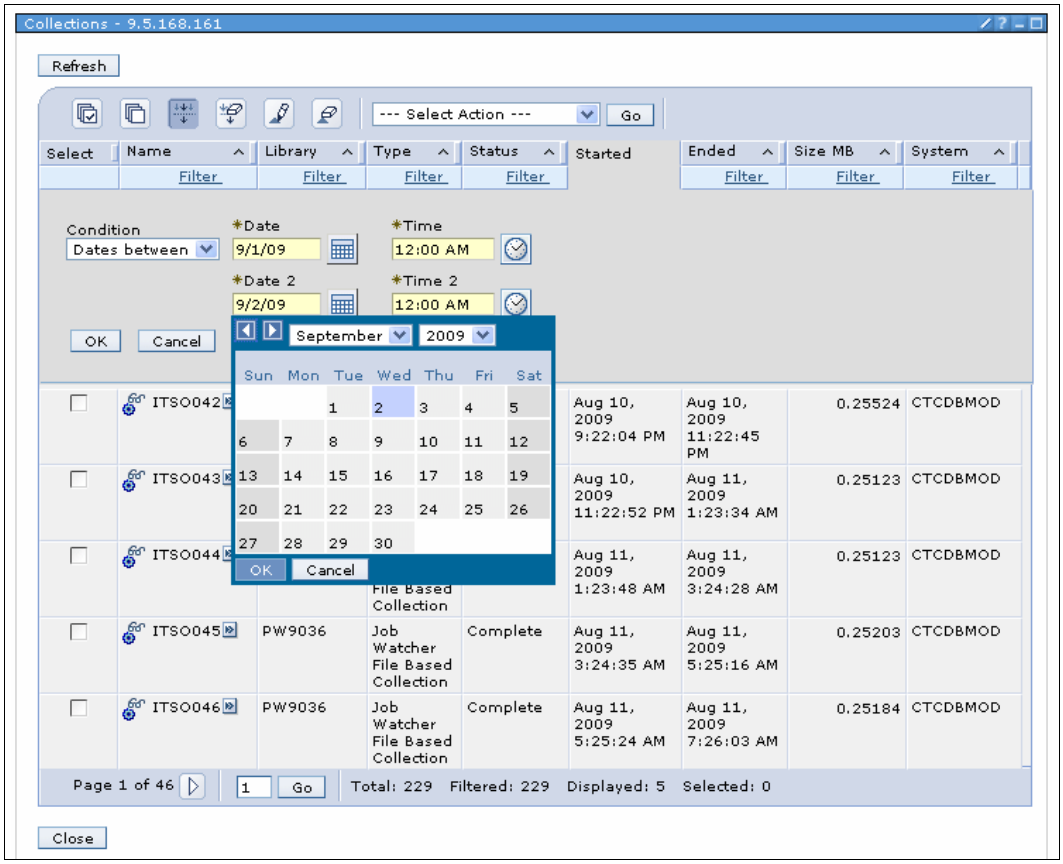


Figure 5-27 Filter collections based on date and time

4. Now that we have filtered all the collections that have started between the date and time we want (1), you can further narrow down the list to show only Job Watcher Collections (2). Now, sort the list by clicking the **Ended** column (3) and choose **Investigate data** (4) from the context menu of the collection, as shown in Figure 5-28.

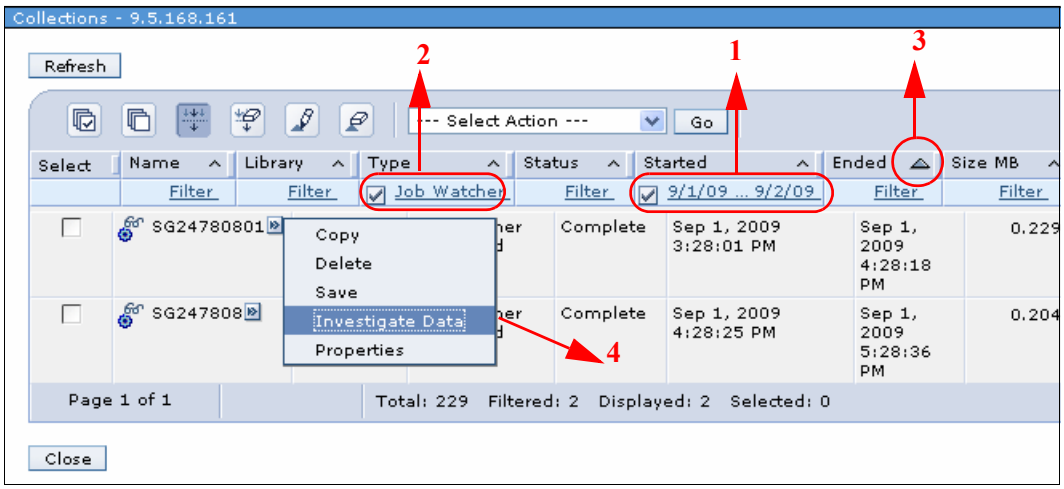


Figure 5-28 Investigate Job Watcher Collection

Step by step analysis using Job Watcher

To perform a step by step analysis of the data using Job Watcher, perform these steps:

1. The default perspective for Job Watcher is Resource Utilization Overview. For our example, because we already know that we want to investigate the database locks, select **CPU Utilization and Waits Overview**, as shown in Figure 5-29.

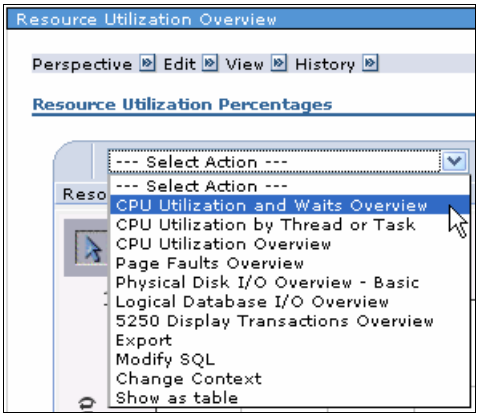


Figure 5-29 Selecting CPU Utilization and Waits Overview

2. Based on our Collection Services investigation earlier, we also know the time when the Database record lock contention occurred. So, you can simply select a few data points within this time frame by clicking the Lock Contention Time area of the bars for further drill-down. Alternatively, you can click **Contention Waits Overview** for all intervals, as shown in Figure 5-30.

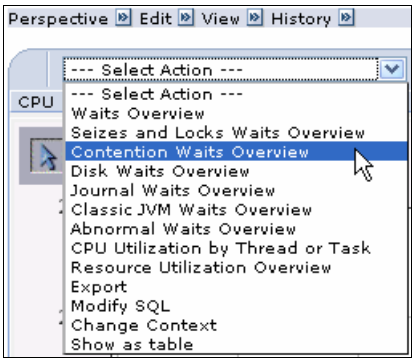


Figure 5-30 Selecting Contention Waits Overview

3. On the Contention Waits Overview chart, use the Zoom Region tool to zoom in on the intervals that show Database Record Lock Contention Time. Using the Select data point tool, click the first interval that shows the Database Record Lock contention time and then select All Waits by Thread or Task sorted by Database Record Lock Contention from the drop-down menu, as shown in Figure 5-31.

> **Note:** Selecting a single interval and then choosing any of the options that show All Waits by Thread or Task sorted by particular metric will be a quicker way to analyze the data. The queries used in the tool to summarize waits will be significantly slower if they have to process all intervals.

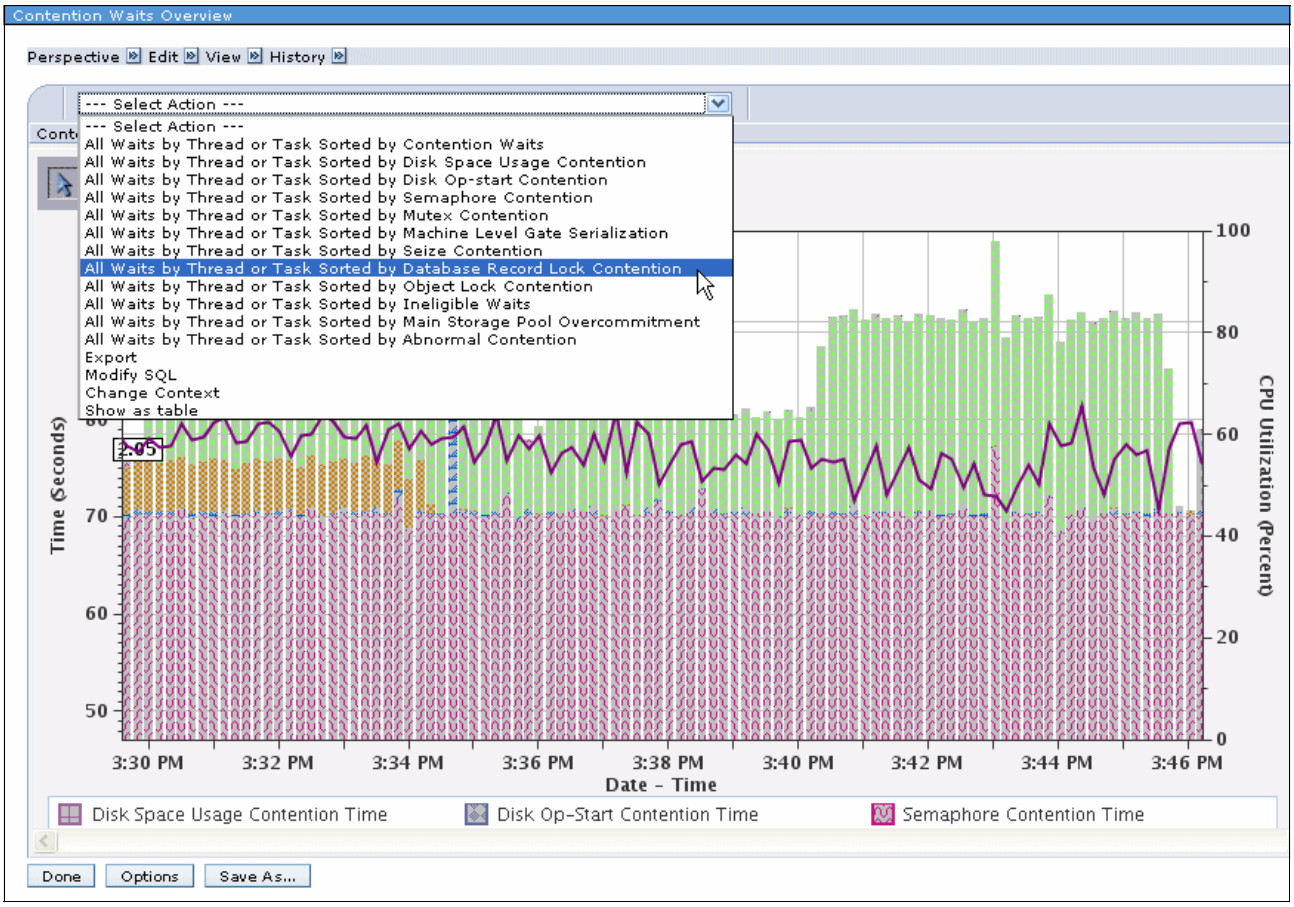


Figure 5-31 Choose All Waits by thread or task sorted by Database Record Lock Contention

4. In the All Waits by Thread or Task sorted by Database Record Lock Contention window shown in Figure 5-32, we see that the job that belongs to user MARK_ROY is at the top. Note that during our investigation of the Collection Services we arrived at two jobs that belonged to the same user but we only see one job right now in Job Watcher. This is because of the way we selected the intervals before the drill-down. When using Job Watcher, we only selected the very first interval that showed Database Record Lock Contention time, while in our analysis of Collection Services, we selected multiple intervals for drill-down.

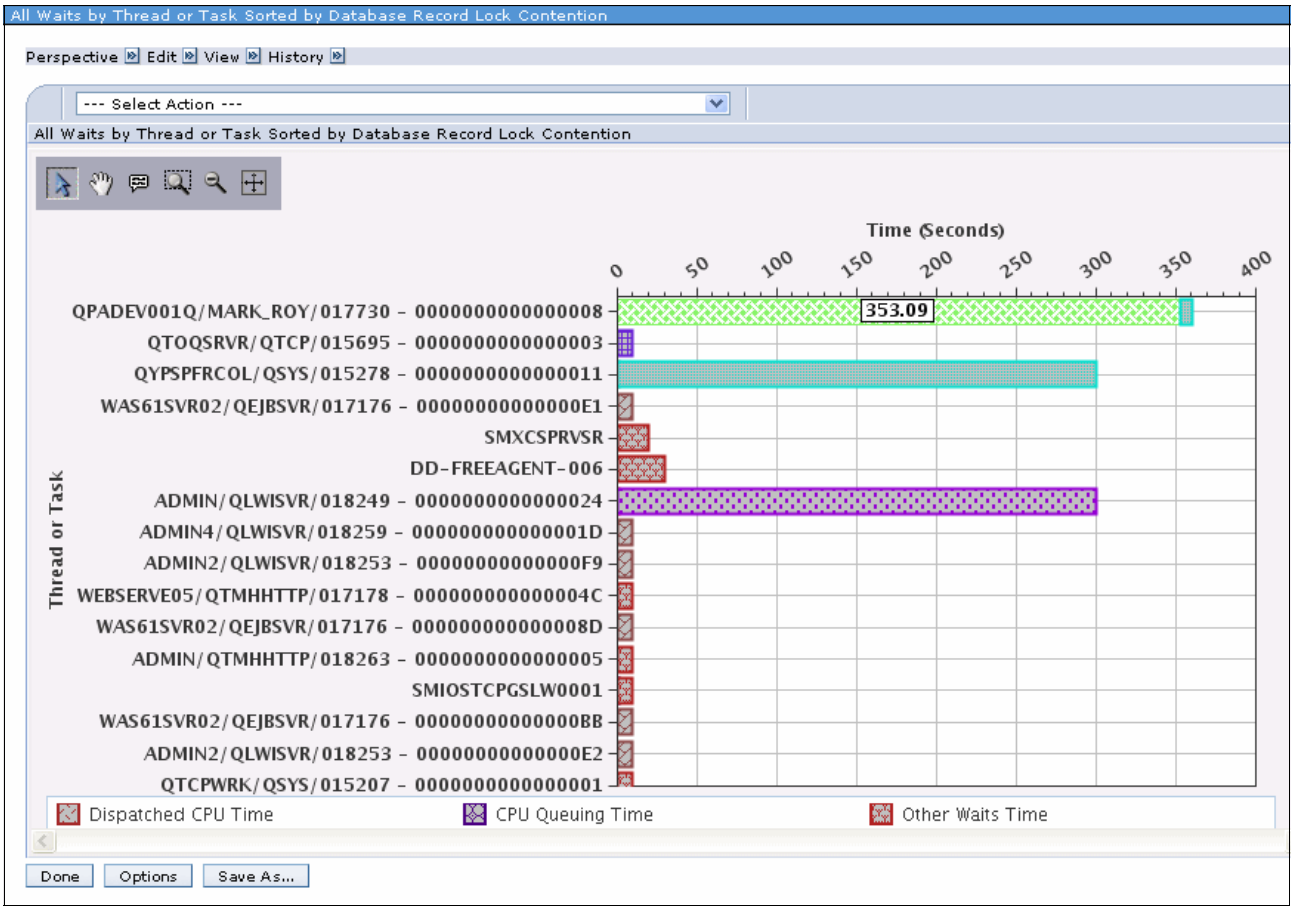


Figure 5-32 Threads or Tasks sorted by Database Record Lock Contention time

5. Now that we have located the job that experienced the problem, we need to determine what caused it. For that task, we need to first look at all waits for this job during the selected intervals. In our case, because we selected only the one interval, the chart shows all waits for that single interval only. Click the job and then select **All Waits for One Thread or Task** from the drop-down menu, as shown in Figure 5-33.

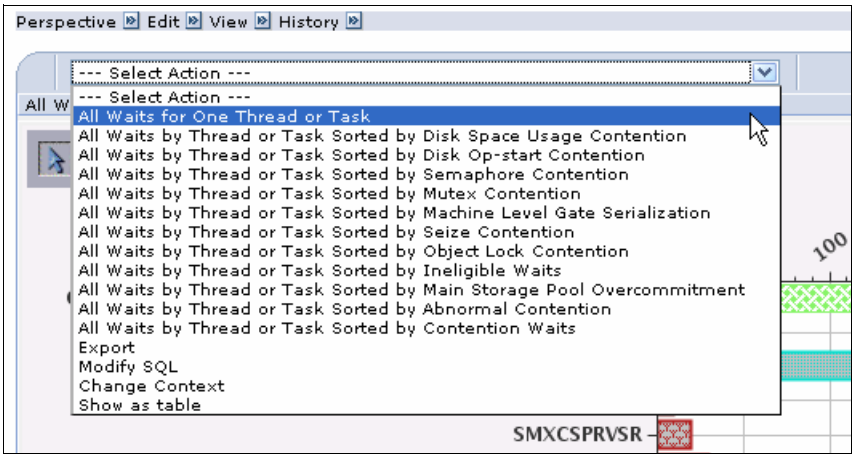


Figure 5-33 Select All Waits for One Thread or Task

6. To determine what this job was doing during this interval and which job is holding the locks, we need to look at the interval details. From the drop-down menu, select **Interval Details for One Thread or Task**, as shown in Figure 5-34.

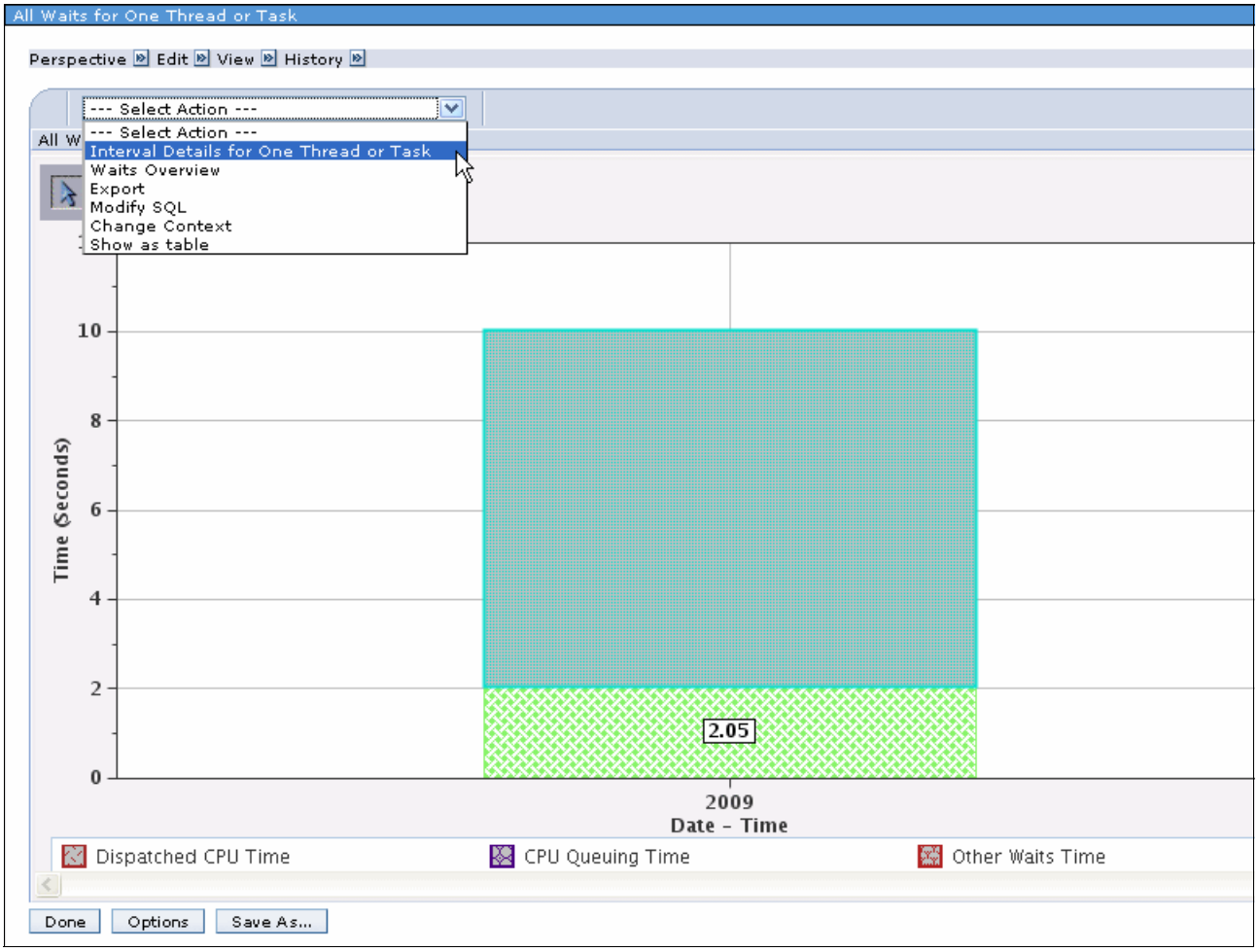


Figure 5-34 Select Interval Details for One Thread or Task

7. The Interval Details for One Thread or Task page has three sections. In the top section, shown in Figure 5-35, we see the Thread or Task Details, including the *Objected Waited on* and the *Holding job or task*. As you can see in the highlighted portion in Figure 5-35, we know that the same user had two jobs running at the same time and both were attempting to update the same database record. If it is a common practice to start multiple sessions while working with the same application, then it may be a good idea to modify the program to alert the user about the record lock. For that task, we need to determine the program that was using this object in question.

Note: If SQL details have been collected, the SQL statement run during this interval will also be displayed. If not, that section is hidden.

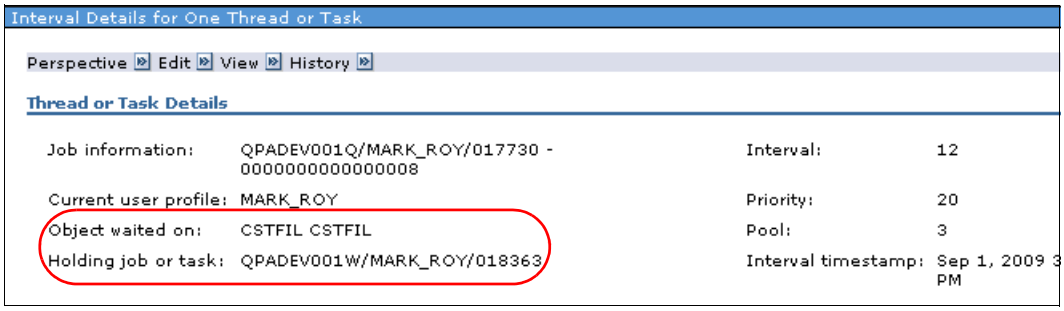
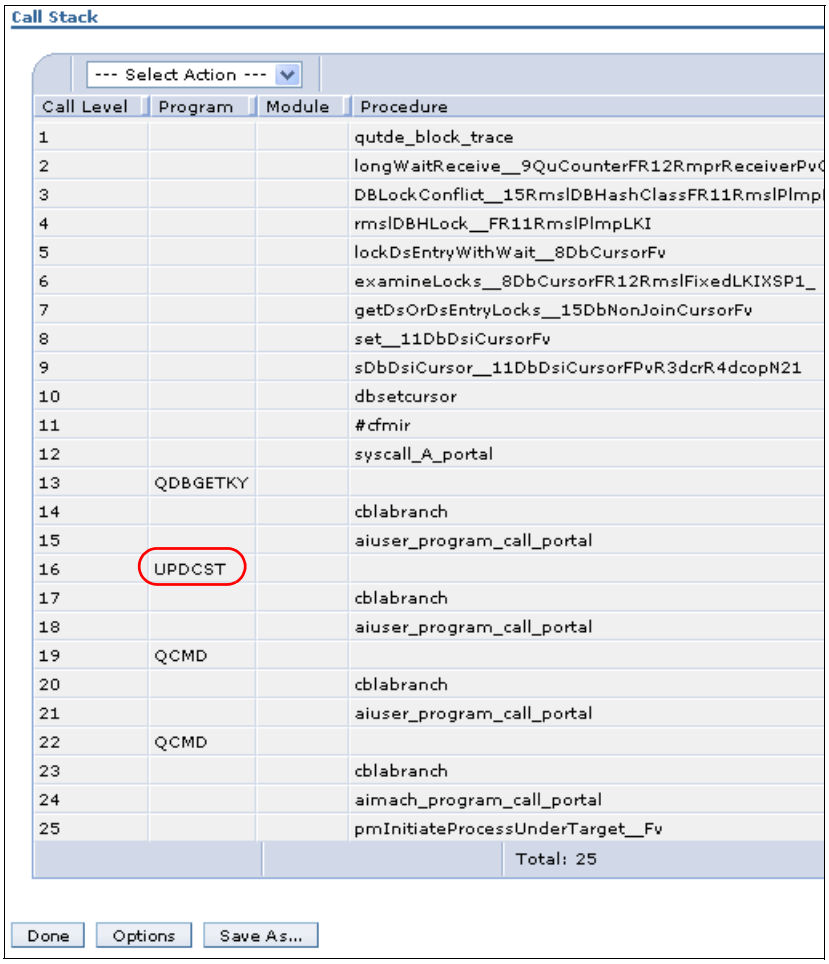


Figure 5-35 Thread or Task Details for an Interval

8. In the bottom section of the Interval Details for One Thread or Task page, we have the Call Stack information, which shows the name of the program, as shown in Figure 5-36. You will notice many lower level procedures in the Call Stack and you may have to scroll down to determine the actual application program. Alternatively, you can also click **Options** and set the number of rows to 25 to display all Call Stack entries.



Call Stack

--- Select Action ---

Call Level	Program	Module	Procedure
1			qutde_block_trace
2			longWaitReceive__9QuCounterFR12RmprReceiverPvC
3			DBLockConflict__15RmslDBHashClassFR11RmslPlmp
4			rmslDBHLock__FR11RmslPlmpLKI
5			lockDsEntryWithWait__8DbCursorFv
6			examineLocks__8DbCursorFR12RmslFixedLKIXSP1_
7			getDsOrDsEntryLocks__15DbNonJoinCursorFv
8			set__11DbDsiCursorFv
9			sDbDsiCursor__11DbDsiCursorFPvR3dcrR4dcopN21
10			dbsetcursor
11			#cfmir
12			syscall_A_portal
13	QDBGETKY		
14			cblabranch
15			aiuser_program_call_portal
16	UPDCST		
17			cblabranch
18			aiuser_program_call_portal
19	QCMD		
20			cblabranch
21			aiuser_program_call_portal
22	QCMD		
23			cblabranch
24			aimach_program_call_portal
25			pmInitiateProcessUnderTarget__Fv
			Total: 25

Done Options Save As...

Figure 5-36 Call Stack information

Alternate analysis using IBM iDoctor for IBM i: Job Watcher

While following our example, you may have noticed that there is more than one way to analyze the same data. As you use the tools provided on IBM i, you may discover that there are often shorter paths and multiple ways to analyze a problem depending on the issue you are working on and how much information you already have.

To illustrate this, let us try an alternate way to analyze the Database Record Lock issue. As you may recall, during our Collection Services investigation we discovered the job name that was experiencing Database Record Lock Contention. For the purposes of this example, let us just use that information to search within the Job Watcher collection:

1. Start IBM iDoctor for IBM i GUI and launch Job Watcher after sign-on.
2. Expand **Job Watcher** → **Libraries** and select the library where the collection resides. In our example, the library is QPFRDATA.

3. In the Collections pane, right-click the collection and select **Search**, as shown in Figure 5-37. In our example, the collection name is SG24780801.

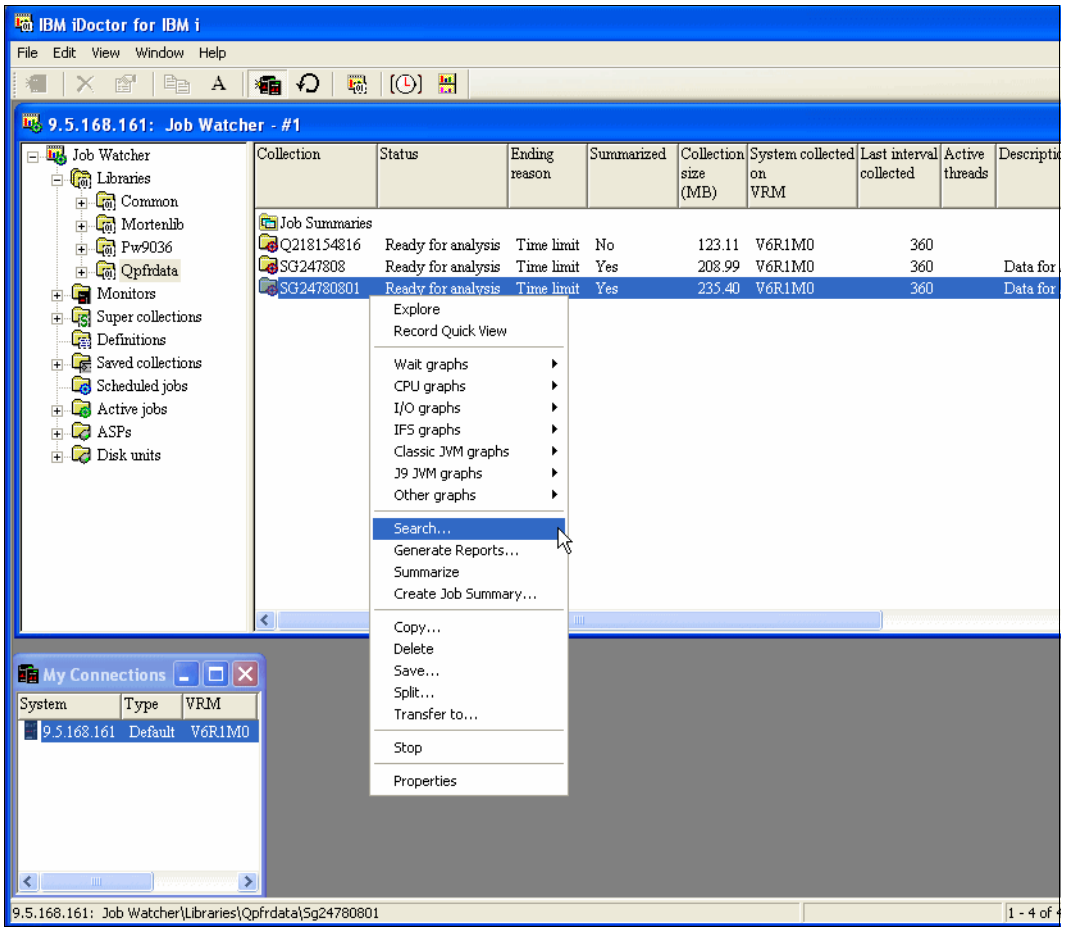


Figure 5-37 Search within a Job Watcher Collection

4. As shown in the Figure 5-38, click **Browse** to open a window with the names of all the jobs within the collection. Alternatively, you can also search based on Current user profile.

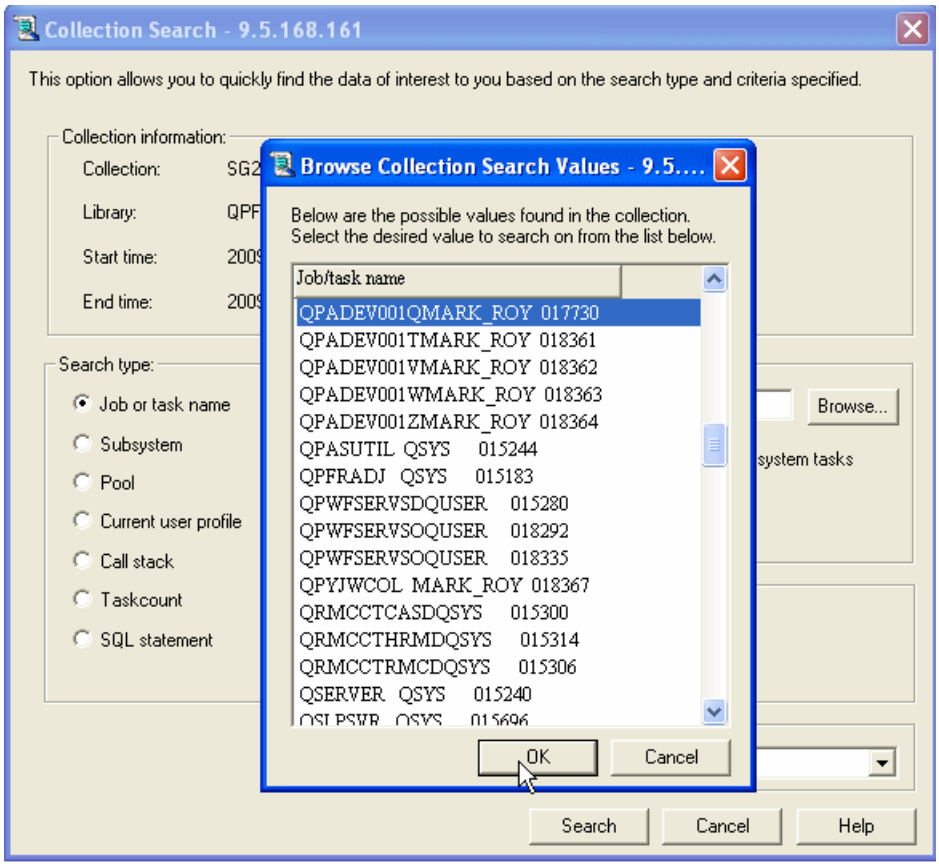


Figure 5-38 Select the job to search for

5. From our Collection Services investigation, we know the job name already. Select that job and click **OK** to go back to the Search window.
6. Click the **Search** button to bring up a table view that shows the life of the job within the selected collection.
7. Click the **Interval number** column to sort the data and scroll right to see various details, including Holding Thread/task name, Wait object name and type, and the program that is being executed in that interval.

8. Alternatively, you can also right-click a row and choose **Thread Wait time signature** to display the life of the thread during the collection time frame, as shown in Figure 5-39.

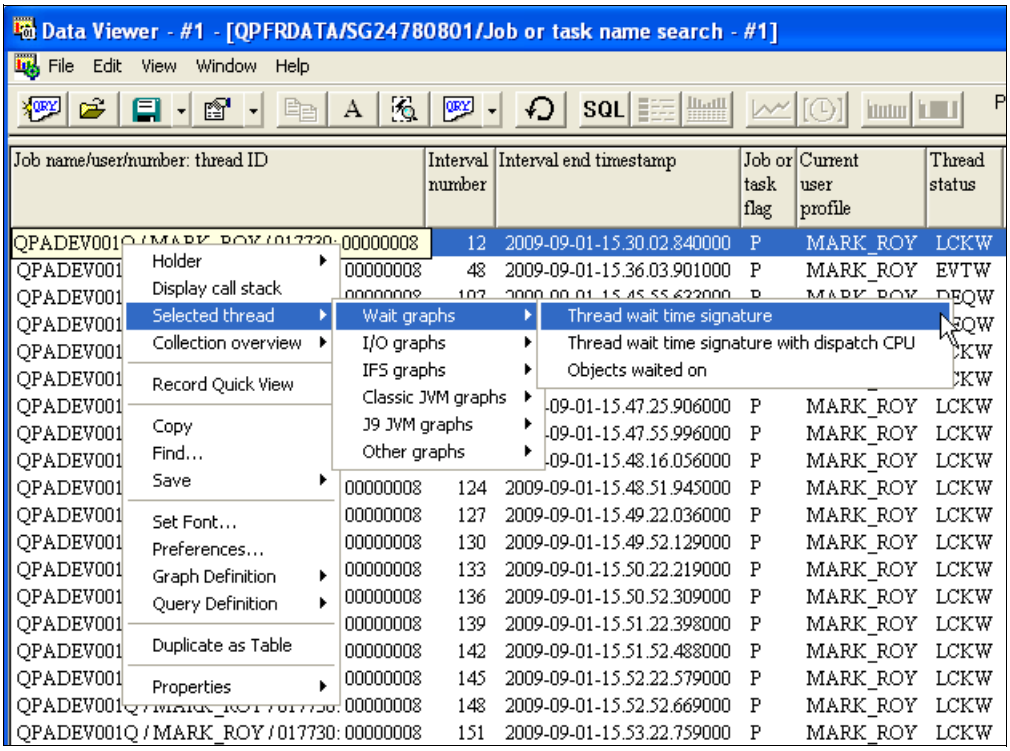


Figure 5-39 Selected Thread Wait time signature

9. In the Thread wait time signature window shown in Figure 5-40, we see that the job continued to wait for a lock for several intervals. The presence of the caret symbol for a particular interval indicates that Job Watcher was able to capture the lock holder information.

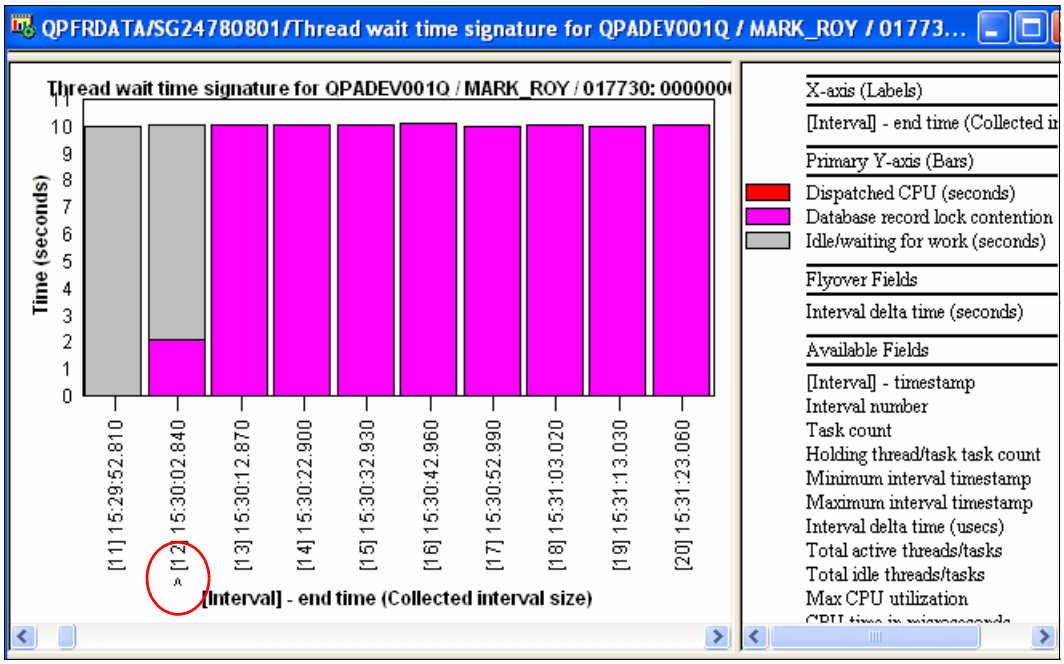


Figure 5-40 Thread wait time signature window

10. You can either double-click the window to bring up the interval details or right-click and select **Display call stack**. As shown in Figure 5-41, the call stack window displays Object Waited on, which includes the name of the file as well as the record number. You can also see the application program name and the holding job or task.

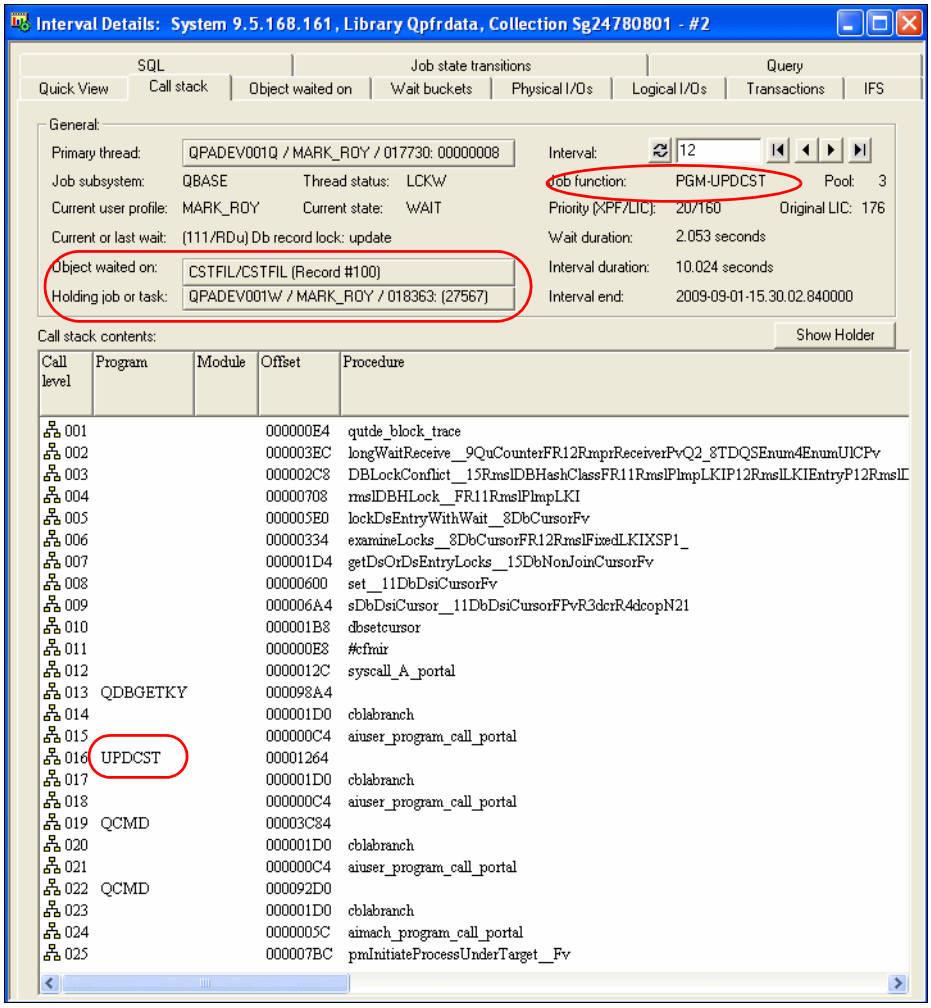


Figure 5-41 Interval Details; Call stack window

5.3.3 Investigate using Disk Watcher data

In our previous example, our particular performance problem was not disk related, so there was no need to use the Disk Watcher performance data investigator. For illustrative purposes, we developed another fictional business problem, generated a workload for it, and collected the corresponding performance metrics. We use this data to discuss the Performance Data Investigator interface, Disk Watcher.

In our fictional business operation, our computer operations team reports that two of the disk drives appear to have a very busy rate, when compared to the other disks on the partition. They do not report any untoward performance issues, but thought the situation should be investigated to determine whether it is, or could become, an issue. They say that they observe high percentage utilization while using the Work with Disk Status (WRKDSKSTS) command. They also provide an example of the output, as shown in Figure 5-42.

```
                              Work with Disk Status                      CTCDBMOD
                                                                 09/11/09  07:43:32
 Elapsed time:   00:07:32

                 Size    %     I/O   Request   Read  Write   Read  Write    %
 Unit  Type       (M)  Used    Rqs  Size (K)    Rqs   Rqs    (K)    (K)  Busy
    1  4328   123488  11.3  632.4      4.0     8.6  623.8    4.7    4.0    16
    2  4328   123488  11.3  603.4      4.0      .1  603.2   29.3    4.0    15
    3  4328   123488  11.3  690.6     10.1     1.0  689.5    8.0   10.1    20
    4  4328   123488  11.3  601.8      4.0      .2  601.6   25.1    4.0    15
    5  4328   123488  11.3  612.2      4.0      .1  612.1   26.4    4.0    15
    6  4328   123488  11.3  577.7      4.1     8.9  568.8    4.7    4.1    14
    7  4328   123488  11.3  628.6      4.0     3.1  625.5    4.4    4.0    16
    8  4328   123488  11.3  590.9      4.0      .4  590.5    8.5    4.0    15
    9  4328   105847  11.3  587.6      4.0      .1  587.5   25.5    4.0    15
   10  4328   105847  11.3  +++++      4.0      .3  +++++   19.6    4.0    77
   11  4328   105847  11.3  +++++      4.0      .1  +++++   22.2    4.0    87
   12  4328   105847  11.3  582.8      4.0      .1  582.7   21.0    4.0    15
```

Figure 5-42 Work with Disk Status

Looking at this output, we can clearly see that disk units 10 and 11 have very high percentage busy rates when compared to the others. Further investigation is warranted.

The first questions that spring to mind are:

- What objects on these two disks are being accessed?
- What programs are accessing these objects?

We know from our experience that Collection Services and Job Watcher do not collect the level of detail we need to answer these questions. Disk Watcher would be the most appropriate tool.

Disk Watcher is very granular in its approach to performance data collection, so we start at the highest level by collecting statistical data. IBM provides six default definitions, which can be used when starting Disk Watcher:

QSTATS All ASPs/disks/pools, dynamic intervals

QSTATS1MIN All ASPs/disks/pools, 1 minute intervals

QTRC All ASPs/disks/pools, dynamic intervals

QTRC1MIN All ASPs/disks/pools, 1 minute intervals

QFULL All ASPs/disks/pools, dynamic intervals

QFULL1MIN All ASPs/disks/pools, 1 minute intervals

A collection using one of the STATS definitions summarizes the I/O data and places it into a set of counters based on the type of I/O operation. A statistics collection always includes all I/O operations on the system. This means that the I/O operations included in the statistics data will not be limited by values specified for the Collection condition (COLCOND) or Collection range (COLRNG) parameters.

A collection using one of the TRC definitions means that a trace collection will be performed. This type of collection gathers data for every I/O that occurs on the specified ASP or disk unit(s).

A collection using one of the FULL definitions means both statistics and trace data will be collected.

We start Disk Watcher using the QSTATS1MIN definition, with a duration of five minutes, at a time we know that the two disks would be busy (computer operations reported that it was happening each day at around the same time). For more information about starting Disk Watcher collections, refer to 4.4, “Disk Watcher” on page 62.

When the collection finishes, we select the Disk Statistical Overview perspective in Disk Watcher under IBM Systems Director Navigator for i, as shown in Figure 5-43.

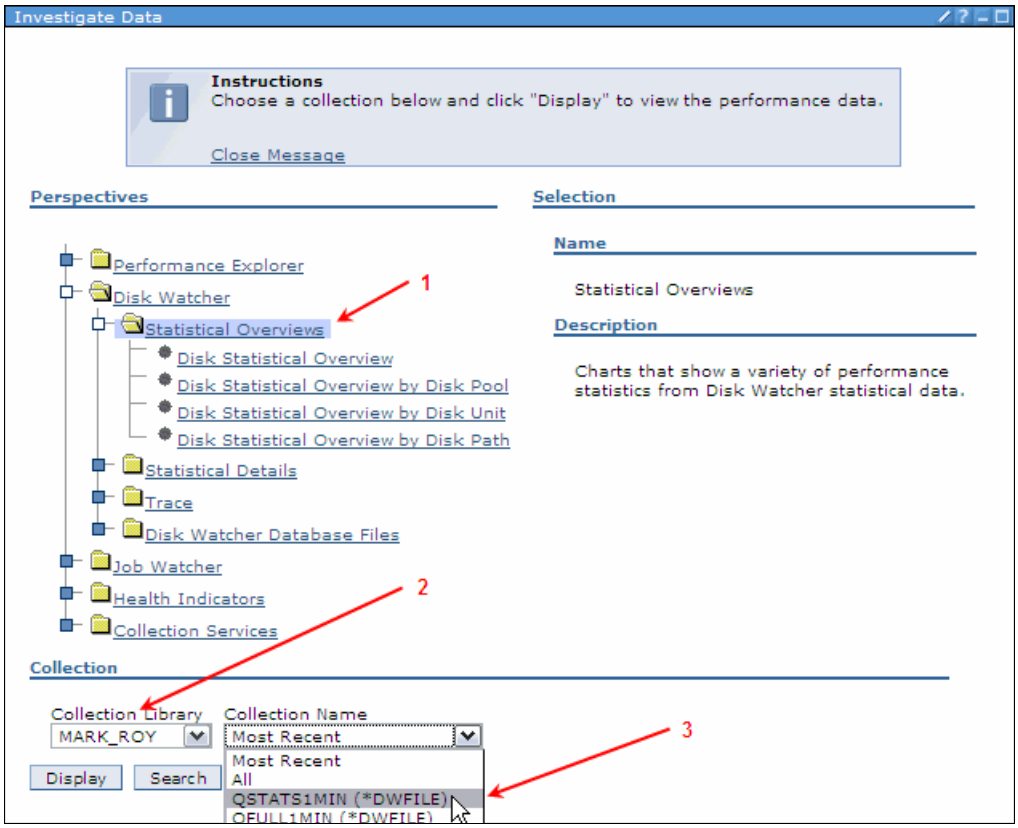


Figure 5-43 Selecting Disk Statistical Overview

In this window, we perform the following steps:

1. Select the perspective. Note that we actually select the Statistics Overview folder and not the Disk Statistical Overview perspective. The Disk Statistical Overview perspective appears because it is the default perspective for the content package.

Note: The top level folder is a “content package”. All sub-folders are simply sub-folders within the content package. Content packages and sub-folders therefore have very similar characteristics, but they are not the same.

2. Select the library containing the collections.
3. Use the Collection Name drop-down menu to select the collection in which we are interested.

We select **Display** to display the perspective. The results are shown in Figure 5-44.

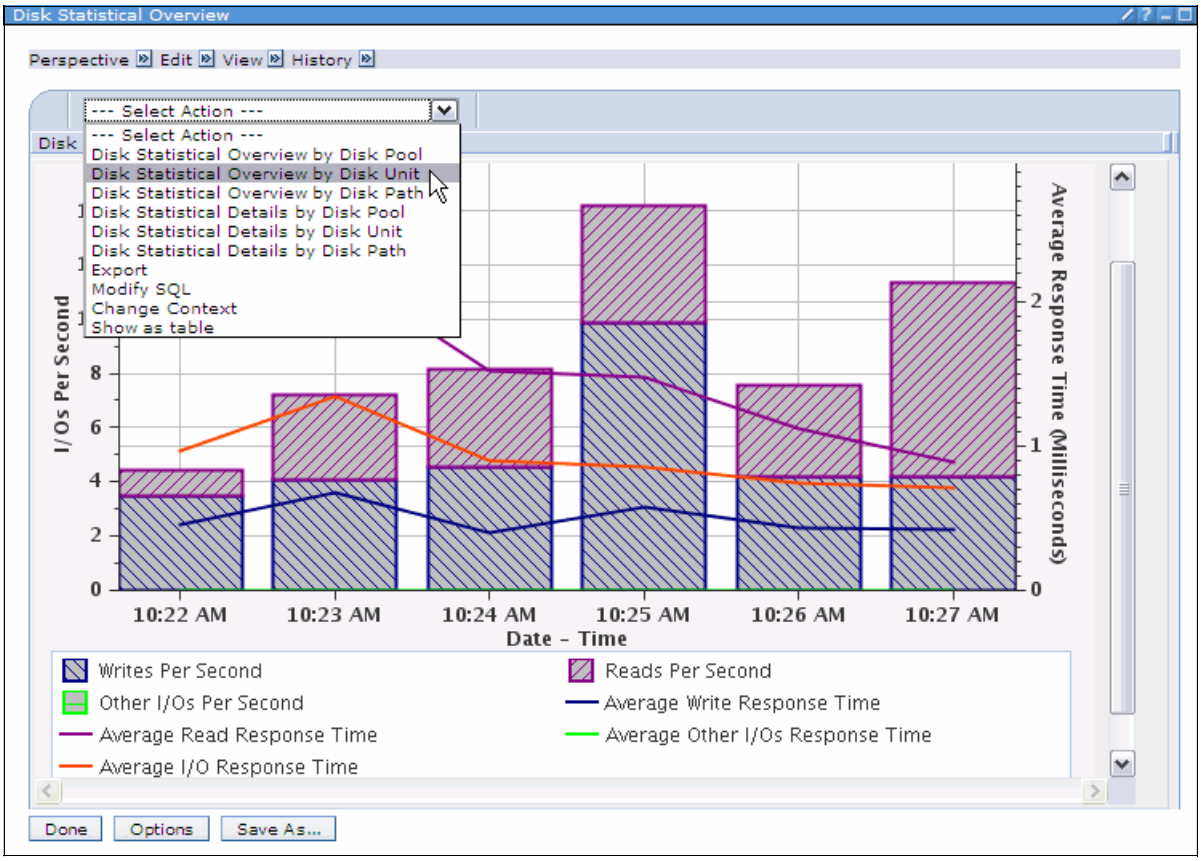


Figure 5-44 Disk Statistical Overview

The Disk Statistical Overview provides some interesting information. We observe that the average disk response time (reads and writes combined) are typically around 1 millisecond. There does appear to be some erratic read response times in the initial period of the collection. Does that indicate a problem? Probably not. Average disk response time around 1 millisecond is good. To confirm whether or not it is a problem, we need to look at the average disk response time over a longer period of time.

Does this graph help us answer the original question of what object and what program was causing the high disk percentage busy? No. We need to drill down further. Next, we select the **Disk Statistical Overview by Disk Unit** perspective from the drop-down menu, as shown in Figure 5-44.

We are presented with two graphs, as shown in Figure 5-45.

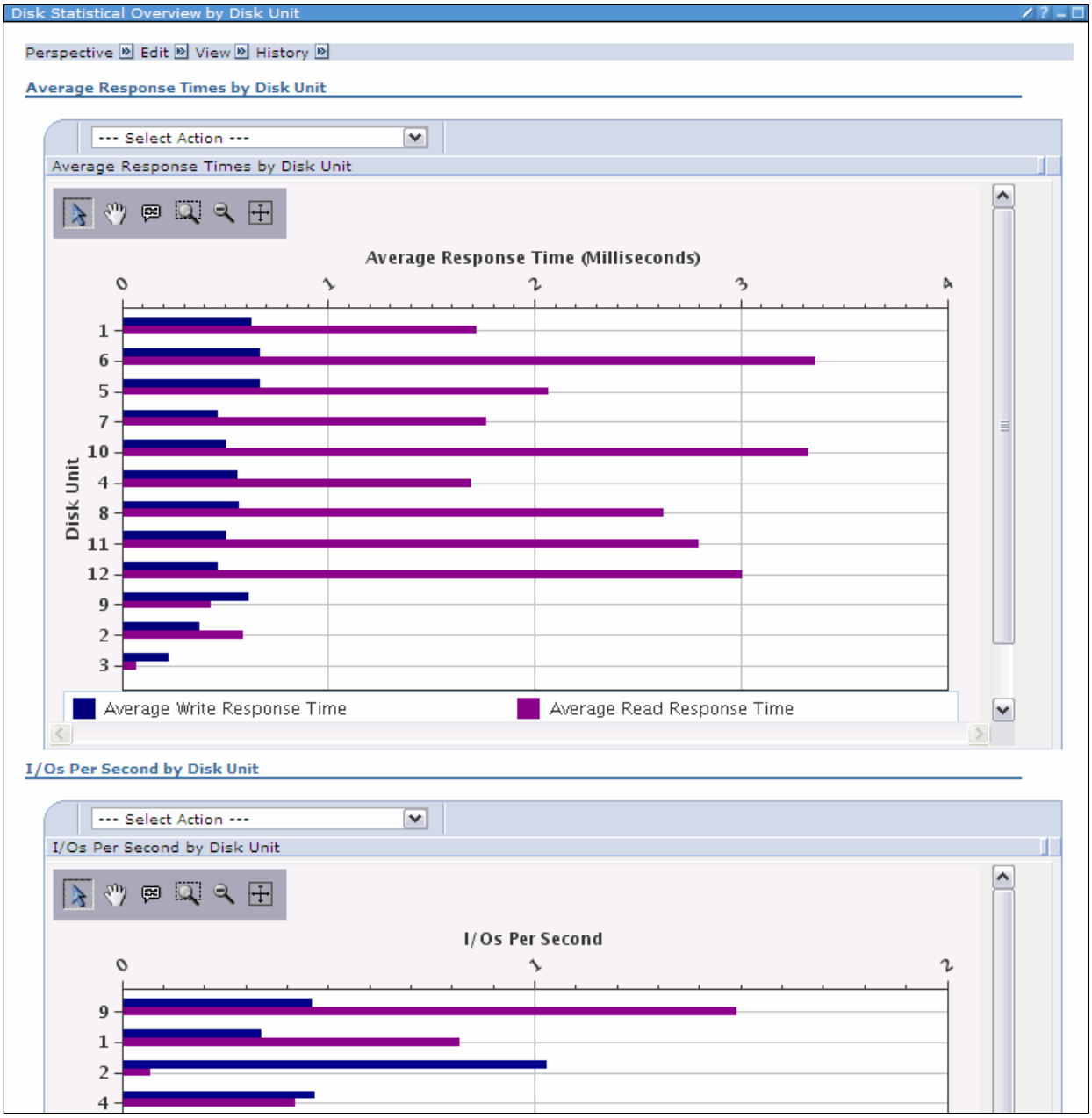


Figure 5-45 Disk Statistical Overview by Disk Unit

> **Note:** With the default Use Patterns check box selected, we find it difficult to distinguish between the bars on the chart. To overcome this problem, we select the **Options** button and clear the check box for **Use Patterns**. That replaces the patterned bars with solid bars.

In the Average Response Times (Milliseconds) graph shown in Figure 5-45, we can see that while the Average Write Response times are fairly consistent between disks, the Average Read Response times do vary significantly. Does this help us answer our question? No. We can see that the Average Read Response times are high for units 10 and 11, but they are as high for other disks. The reason why some are much higher might warrant further investigation at another time. We need to look elsewhere for our answers.

We bypass the Statistical Details perspectives in this investigation. The perspectives are a window into the underlying data used to generate the Statistical Overview graphs. Given that we did not find what we are looking for in the overviews, it is unlikely the details will provide our answers. We move on to trace data.

We select the **Done** button twice to return to the Investigate Data pane. From there, we select the **I/O Counts by IOP, I/O Type, Pool, Object, Object Type, Module, Procedure** perspective, as shown in Figure 5-46.

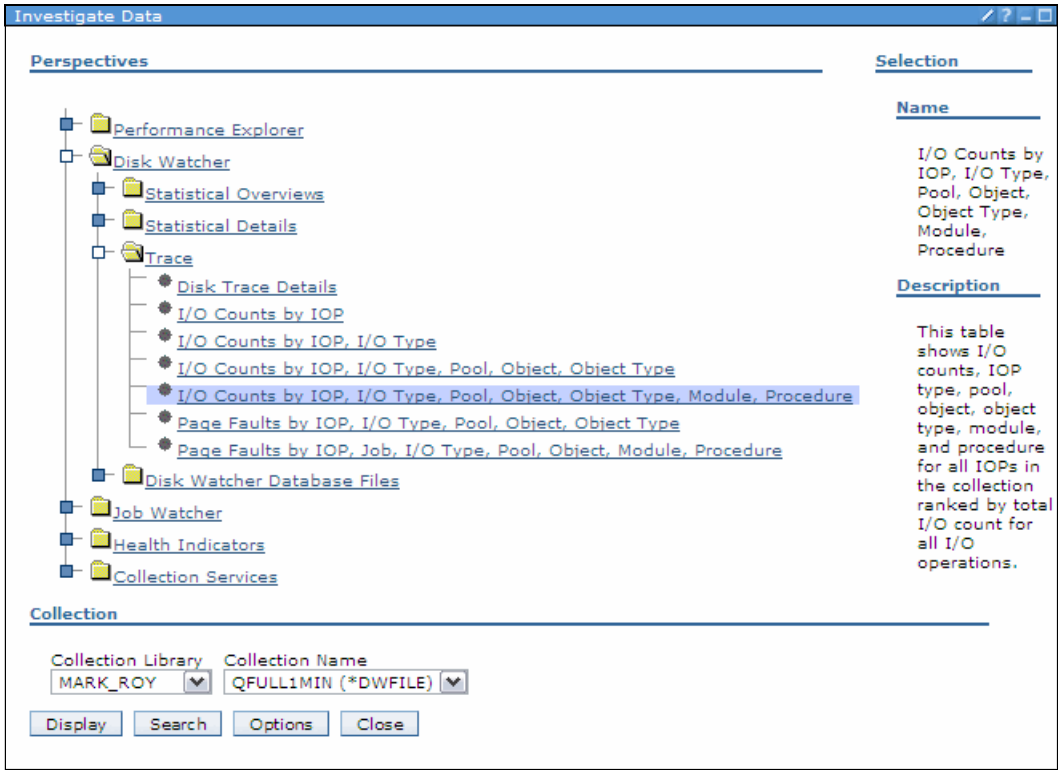


Figure 5-46 Selecting I/O Counts by IOP, I/O Type, Pool, Object, Object Type, Module, Procedure perspective

We select this perspective because the title sounded like it might answer our questions. The results are shown in Figure 5-47.

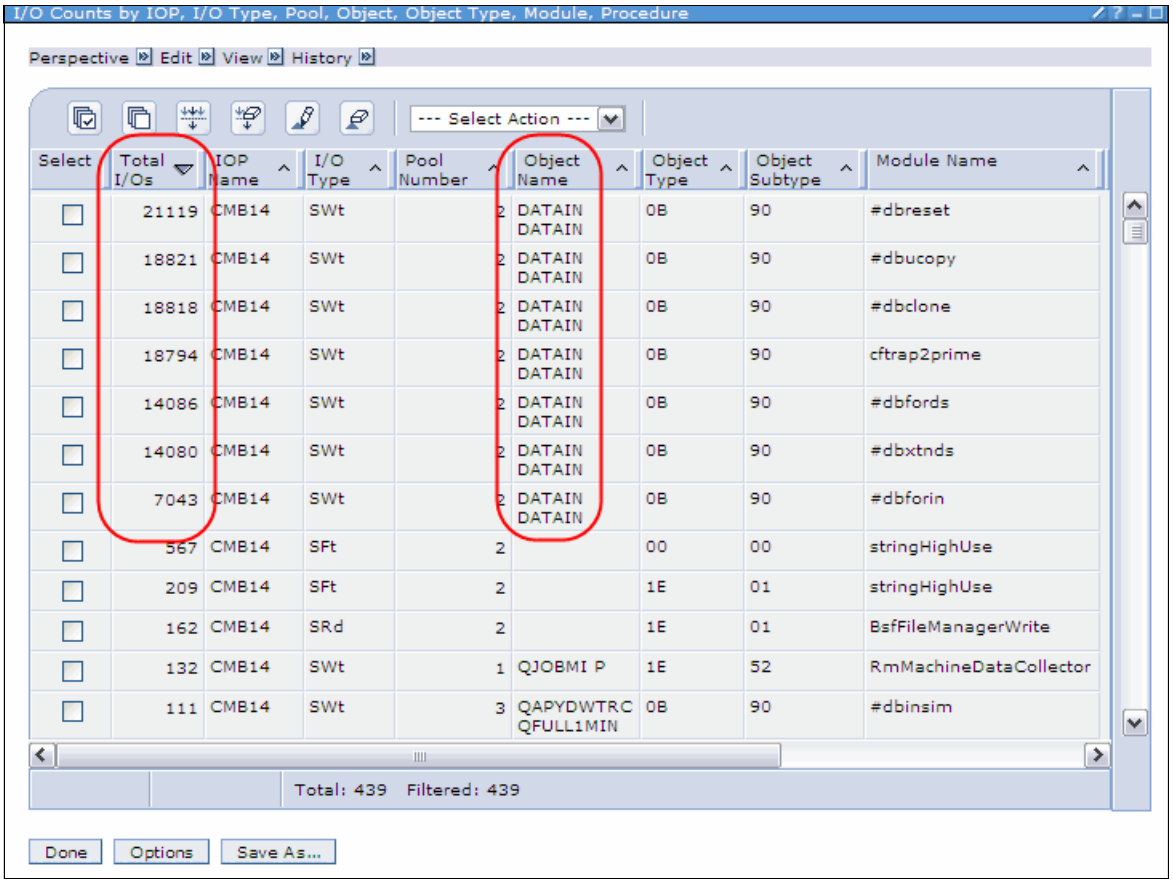


Figure 5-47 I/O Counts by IOP, I/O Type, Pool, Object, Object Type, Module, Procedure perspective

We notice that one object, DATAIN, seems to account for most of the total I/O. We can tell from the I/O Type column that the I/Os were writes. The I/O Types are provided in Table 5-1.

Table 5-1 I/O types

I/O Types	Segment address range fault
SFt	Segment address range fault
SCl	Segment address range clear
SRd	Segment address range read
SWt	Segment address range write
SRv	Segment address range remove
SUp	Segment address range unpin
SWp	Page out task write
STv	Segment address range trivial request
GRf	Access group read
GPg	Access group purge

I/O Types	Segment address range fault
SRP	Segment address range remove request IOP
SCP	Segment address range clear request IOP
GCP	Segment address range clear request POW
SUP	Segment address range unpin request IOP
SRQ	Segment address range read request IOP
GRQ	Segment address range read request POW
SFP	Segment address range fault request IOP
GFP	Segment address range fault request POW
SRR	Access group read IOP
GRR	Access group read POW
SWP	Segment address range write request IOP
GWP	Segment address range write request POW
GPP	Access group purge request IOP
SPw	Segment address range page out wait request

Do we now have the answer to our question? Maybe.

We can see from this perspective that DATAIN is a heavily used object. Unfortunately, the I/O Counts by IOP, I/O Type, Pool, Object, Object Type, Module, Procedure perspective only shows the data aggregated at the IOP level, not the disk unit level. One could make an assumption that this object is the one we are looking for, but that is just an assumption based on the available data.

We hope that the Module or Procedure data in the perspective would show us which application programs are using the file. Unfortunately, we can only see the IBM i module name that the application is indirectly invoking.

So, at least we can start looking at methods, outside of Performance Data Investigator, to determine what is using this file. Scanning source code for the application may be one method. Seeing which jobs have object locks on the object might be another method.

By now you might be wondering why we include this example when we did not satisfactorily answer our questions. Disk Watcher was primarily designed to capture disk I/O activity below the SLIC layer. This data is very useful when investigating mirroring or multi-path I/O performance problems. However, we have also shown that it can be used to identify interesting facts and patterns in all I/O, which may be useful when investigating certain problems using the other tools.

The last feature we describe is the ability to modify the SQL used by Performance Data Investigator to retrieve the data for the graphs. If we select the **Modify SQL** option from the drop-down menu, the window shown in Figure 5-48 opens.

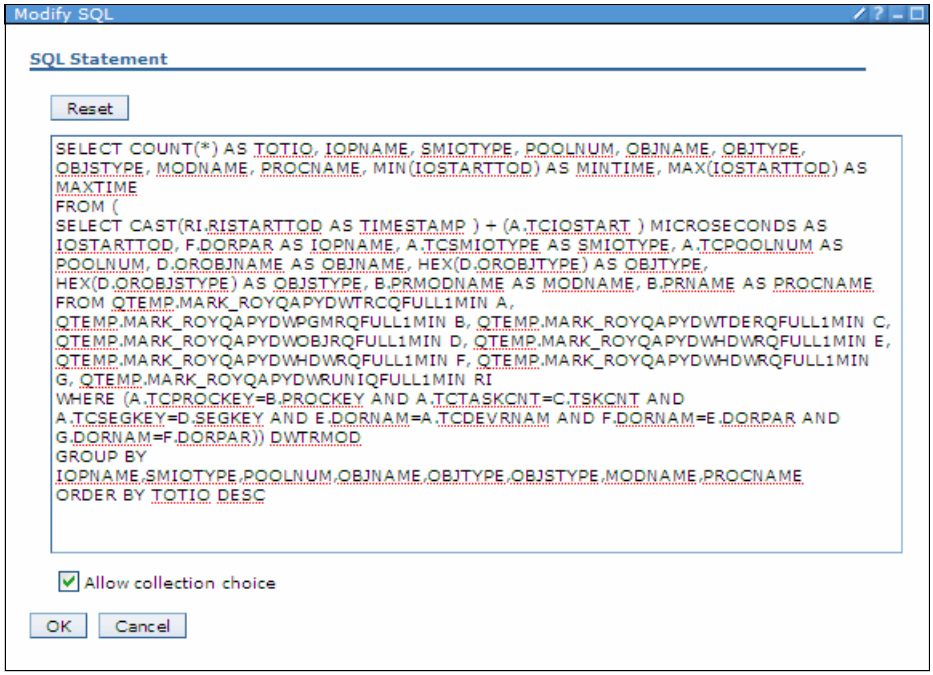


Figure 5-48 Modify SQL

This window shows the IBM supplied SQL statement that was used. It can be used to modify the SQL to retrieve data you may want to see in the perspectives. This can be a powerful option when the IBM supplied perspectives do not provide the information for which you are looking.

Note: An altered perspective can be saved. This means that modified SQL, selected points, and so on, are all saved for future use. This is a very valuable feature after you have drilled down deep into an investigation you might want to come back to later.

6

Real-time monitoring tools

Monitoring allows you to automate the management of your partition and gives you real-time notification about user-defined situations.

In this chapter, we discuss the various tools that are available to monitor IBM i V6.1 in real time. This chapter discusses the following topics:

- Management Central monitors
- CL commands for real-time monitoring
- IBM Systems Director Navigator for i:
  - Performance Tasks for real-time monitoring
  - Health Indicators
- IBM Director
- Tivoli Integration
- Watch for Event function

© Copyright IBM Corp. 2009. All rights reserved.

6.1 Management Central

Management Central is a portion of the function that appears in the IBM Navigator for i GUI. It can manage IBM i systems and partitions. Functions that appear in menus under a system listed in My Connections run on a single system basis, while Management Central functions can be invoked on a single system or to multiple systems in the network. Some of the functions that appear under a My Connections system, such as running a command on a system or collecting inventory information, really invoke a Management Central function to a single system or endpoint.

Using the Management Central function, you can run monitors on one or more endpoints. For system monitors, up to 26 different metrics can be monitored based upon system measurement data gathered by the Collection Services function of IBM i. Jobs, files, and message monitors are also available in Management Central. Using the Graph History capability with system monitors, a user can obtain a historical view of any of the 26 system monitor metrics for one or more endpoints.

6.2 Management Central monitors

Using the Management Central is probably the easiest way to monitor the partition performance.

You can create monitors for your systems, jobs, messages, and files. These monitors allow you to monitor your system performance, job status, message queues, and system log text files across multiple endpoint systems. They provide you with the flexibility you need to access information for the IBM i partitions you support.

6.2.1 Monitor types

Monitors use values to measure (metrics) and values to compare against (thresholds), which are described in the following sections, The current values of the metrics are collected on a schedule you select, compared to the threshold values you define, and then actions are taken based on that comparison.

Job monitors monitor a job or list of jobs based on job name, job user, job type, subsystem, or server type. Metrics, or values to monitor, for job monitors include eight of the 26 System monitor metrics, job status, and messages in the job log for the selected jobs.

Message monitors monitor any message queue for any group of messages that you select.

File monitors monitor selected files or all system logs for a text string, for a specified size, or for any changes to the file. You can also monitor the history log for a specified text string.

System monitors

System monitors let you visualize what is going on with your systems with detailed graphs that monitor the real-time performance of multiple IBM eServer iSeries servers. You can view and change information about the monitor, including the metrics to be collected by the monitor.

Use the Graph History window to see a graphical view of the metrics that have been collected for an extended period of time for a particular monitor. You can contrast this data with the real-time data for the last hour shown in the Monitor window.

Note: Management Central uses Collection Services data for its data. Therefore, tools like PDI or iDoctor can be used later to revisit information originally noticed in System monitors.

The Monitor window (Figure 6-1) shows a graphical view of the metrics for a monitor as they are being collected. You can have more than one monitor window open at the same time, and you can work with other windows while the Monitor windows are open. The Monitor window contains three panes: Graphs, Details, and Properties. A window can also be opened showing the event log (history of monitor threshold trigger and reset events).

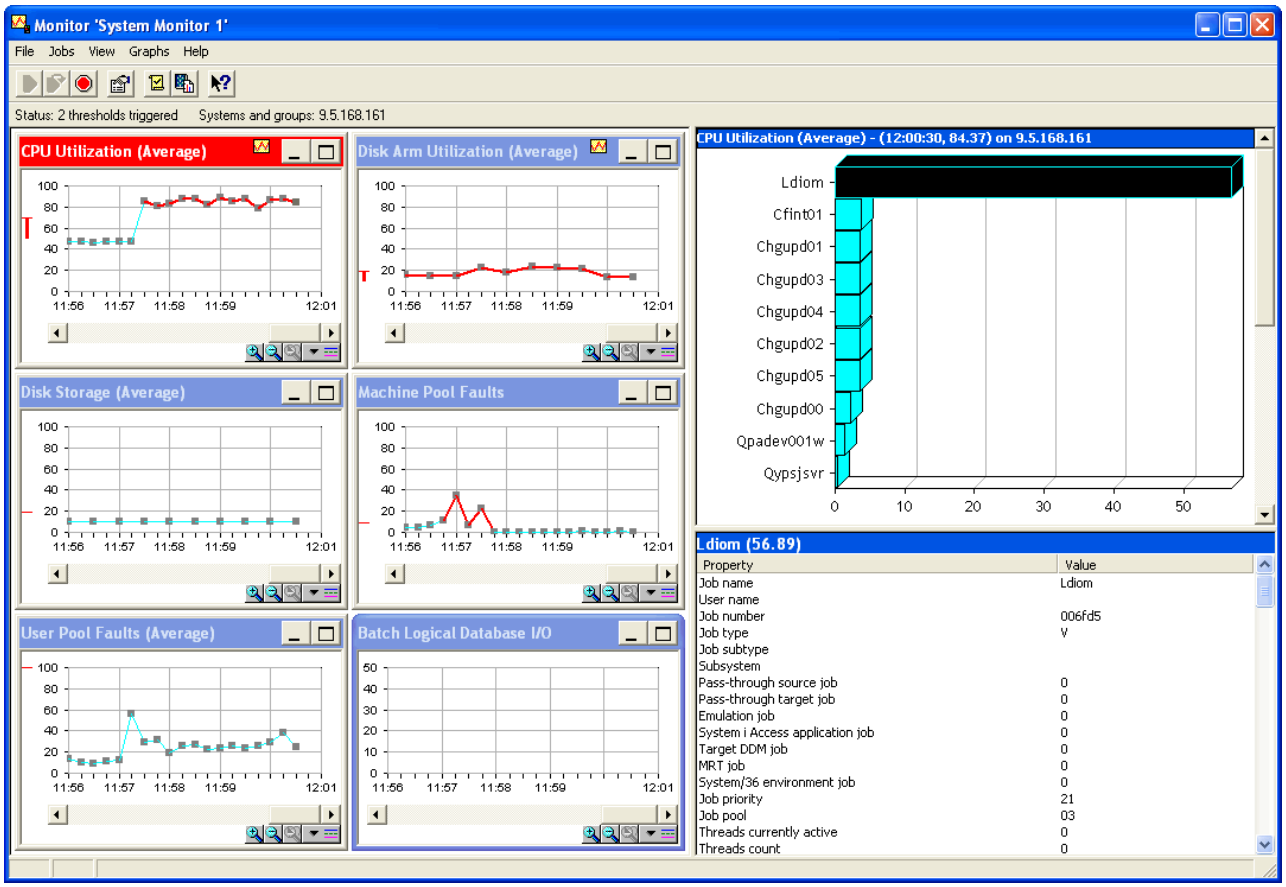
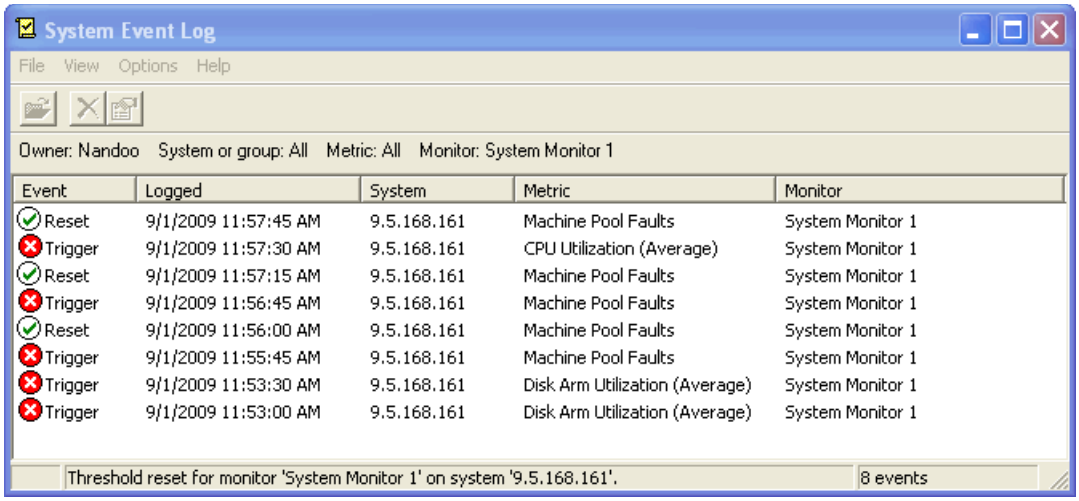


Figure 6-1 System Monitor window

System Event Log

The System Event Log is a list of threshold trigger and reset events for your monitors. You can specify on the Properties page for each monitor whether you want events added to the event log. To see the Properties page for any monitor, select the monitor in the list of monitors and then select **Properties** from the File menu. You can right-click a monitor and choose **Event log** to open a list of all events for that monitor. A sample Event log is show in Figure 6-2.



Event	Logged	System	Metric	Monitor
Reset	9/1/2009 11:57:45 AM	9.5.168.161	Machine Pool Faults	System Monitor 1
Trigger	9/1/2009 11:57:30 AM	9.5.168.161	CPU Utilization (Average)	System Monitor 1
Reset	9/1/2009 11:57:15 AM	9.5.168.161	Machine Pool Faults	System Monitor 1
Trigger	9/1/2009 11:56:45 AM	9.5.168.161	Machine Pool Faults	System Monitor 1
Reset	9/1/2009 11:56:00 AM	9.5.168.161	Machine Pool Faults	System Monitor 1
Trigger	9/1/2009 11:55:45 AM	9.5.168.161	Machine Pool Faults	System Monitor 1
Trigger	9/1/2009 11:53:30 AM	9.5.168.161	Disk Arm Utilization (Average)	System Monitor 1
Trigger	9/1/2009 11:53:00 AM	9.5.168.161	Disk Arm Utilization (Average)	System Monitor 1

Figure 6-2 System Event Log

By default, the list of events is arranged in order by date and time. To change the order, you can click any column heading. For example, to sort the list by the endpoint system where the event occurred, click **System**.

You can customize the list of events to include only those that meet specific criteria by selecting **Options** from the menu bar and then selecting **Include**.

You can have more than one Event Log window open at the same time, and you can work with other windows while the Event Log windows are open. Event Log windows are updated continuously as events occur.

File monitors

You can use a file monitor to monitor all system logs, or you can select a set of specific files to monitor. You can monitor for a specified size or for specific text strings.

Job monitors

You can use a job monitor to monitor a job or list of jobs based on job name, job user, job type, subsystem, or server type. The metrics for job monitors include eight performance values, job status, and messages in the job log. For example, you might want to monitor a job's CPU usage, job status, and job log messages. The job monitor allows you to define commands to run when a specified threshold is met.

Message monitors

You can create a message monitor to monitor any message queue for any group of messages you select. The monitor can run on one or more systems. For example, you might monitor the system operator message queue on a group of systems for message ID CPI0953, which is sent when the threshold of a disk pool has been exceeded. The message monitor also allows you to specify a command to run on the server when the message count exceeds a specified level. So when the message monitor detects CPI0953, and the threshold of a disk pool is

exceeded, you could specify to run a command that would delete objects that you no longer need from the disk pool.

For each message queue, you can specify two separate and independent message sets to monitor. For each message set, you can specify the threshold (number of messages received in the queue) that will trigger an event and an operating system command that will run when the event is triggered.

You can also see the details of a message, reply to a message, send a message, delete a message, and see or change the properties of a message.

B2B activity monitors

Business to Business (B2B) monitors provide some specific metrics relating to WebSphere transactions. You can use a B2B activity monitor to view a graph of active transactions over time, and you can run commands automatically when thresholds are triggered. You can search for and display a specific transaction as well as view a bar graph of the detailed steps of that specific transaction.

6.2.2 Metrics

A metric is a measurement of a particular characteristic of a system resource or the performance of a program or a system. You can specify the metrics to be monitored by a system monitor, job monitor, message monitor, file monitor, or a B2B activity monitor.

For a system monitor, you can select from a wide range of available metrics (26 metrics), such as CPU utilization, interactive response time, transaction rate, disk arm utilization, disk storage, disk IOP utilization, and more. Please note that some metrics have different collection intervals from the other metrics. A good example of this is the CPU that can be measured every 15 seconds, whereas disk utilization is only measured in 30 second intervals.

Additionally, there are metrics that appear similar but provide a different level of detail. For example, *CPU Utilization Basic* and other *CPU utilization* metrics have the same default interval levels, but offer different levels of detail. Basic is a very low impact method of obtaining CPU utilization and can be run on any system at 15 second intervals without any impact. The other CPU Utilization metrics collect data on every job/task/thread on the system for L2 and L3 drill-down. These CPU utilization metrics consume man more resources depending upon the number of tasks and threads running in the system and results in much more data going into the *MGTCOL every interval. There might even be missing intervals if the system cannot keep up with the collection of data at this level of detail.

For a job monitor, available metrics include job count, job status, job log messages, CPU utilization, logical I/O rate, disk I/O rate, communications I/O rate, transaction rate, and more.

For a message monitor, you can specify one or more message IDs, message types, and severity levels. You can also select from a list of predefined sets of messages that would be associated with a specific type of problem, such as a communications link problem, a cabling or hardware problem, or a modem problem.

For a file monitor, you can select to monitor files across multiple endpoint systems for specified text strings or for a specified size. Also, you can select to trigger an event whenever a specified file has been modified.

For a detailed description of each metric in the Management Central Monitors, refer to IBM i 6.1 Information Center and select **Connecting to System i** → **System i Navigator** → **Management Central** → **Working with Management Central monitors** → **Creating a new monitor Selecting the metrics**.

6.2.3 Thresholds

A threshold is a setting for a metric that is being collected by a monitor. This setting allows you to specify a trigger value, which, when exceeded, triggers an event. Actions can then be taken when the trigger value is reached. You can also specify actions to be taken when a second value, called the reset value, is reached. For example, when you create a system monitor, you can specify an operating system command that stops any new jobs from starting when CPU utilization reaches 90 percent, and another operating system command that allows new jobs to start when CPU utilization falls to less than 70 percent.

For some metrics, it is appropriate to specify a reset value, which resets the threshold and allows it to be triggered again when the trigger value is reached. For those thresholds, you can specify a command to be run when the reset value is reached. For other metrics (such as the File Status metric and the Text metric on file monitors, and any message set on a message monitor), you can specify to automatically reset the threshold when the trigger command is run.

You can set up to two thresholds for each metric that the monitor is collecting. Thresholds are triggered and reset based on the value at the time the metric collection is made. Specifying a higher number of collection intervals in the Duration field helps avoid unnecessary threshold activity due to frequent spiking of values.

You can choose to add an event to the event log whenever the trigger value or the reset value is reached. You can also use the trigger to display the event log.

Using threshold commands

Threshold settings are used to automate any IBM i command you want to run when thresholds are triggered or reset. For example, suppose you are running a job monitor and a certain batch job that is supposed to complete before the first shift begins is still running at 6:00 a.m. To accomplish this task, you can set up Threshold 1 to send a page command to a system operator to look at it. You can also set up Threshold 2 to send a command to end the job if it is still running at 7:00 a.m.

In another situation, you might want to notify your operators with a page command when the job monitor detects that the wait time values for the FTP and HTTP servers have reached a median level. If the FTP server jobs end, you can restart the server with a start server command (such as STRTCPSVR *FTP). You can set thresholds and specify commands to automatically handle many different situations. In short, you can use threshold commands in any way that makes sense for your environment.

Management Central monitors allow you to specify any batch commands to run on the server when the threshold is triggered or reset. You can enter an IBM i command name and click **Prompt** (or press F4) for assistance in specifying the parameters for the command. You can even use replacement variables (such as &TIME or &NUMCURRENT) to pass information to the command, such as the time and actual value of the metric.

6.2.4 Configuring and starting a System Monitor

To configure and start a system monitor, perform the following steps:

1. Start the IBM Navigator for i, click **Management Central** and, if needed, sign on to the partition used as the central system.
2. Select **Monitors** → **System** → **New Monitor**, as shown in Figure 6-3.

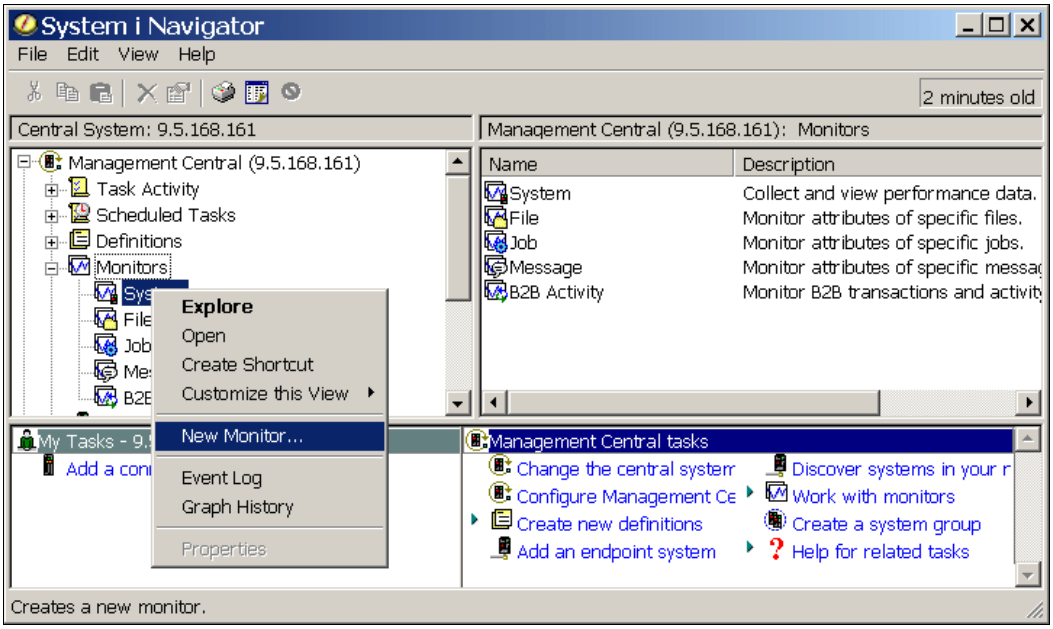


Figure 6-3 Starting the new monitor

3. Specify the name and enter a meaningful description of the monitor in the **General** tab, as shown in Figure 6-4.

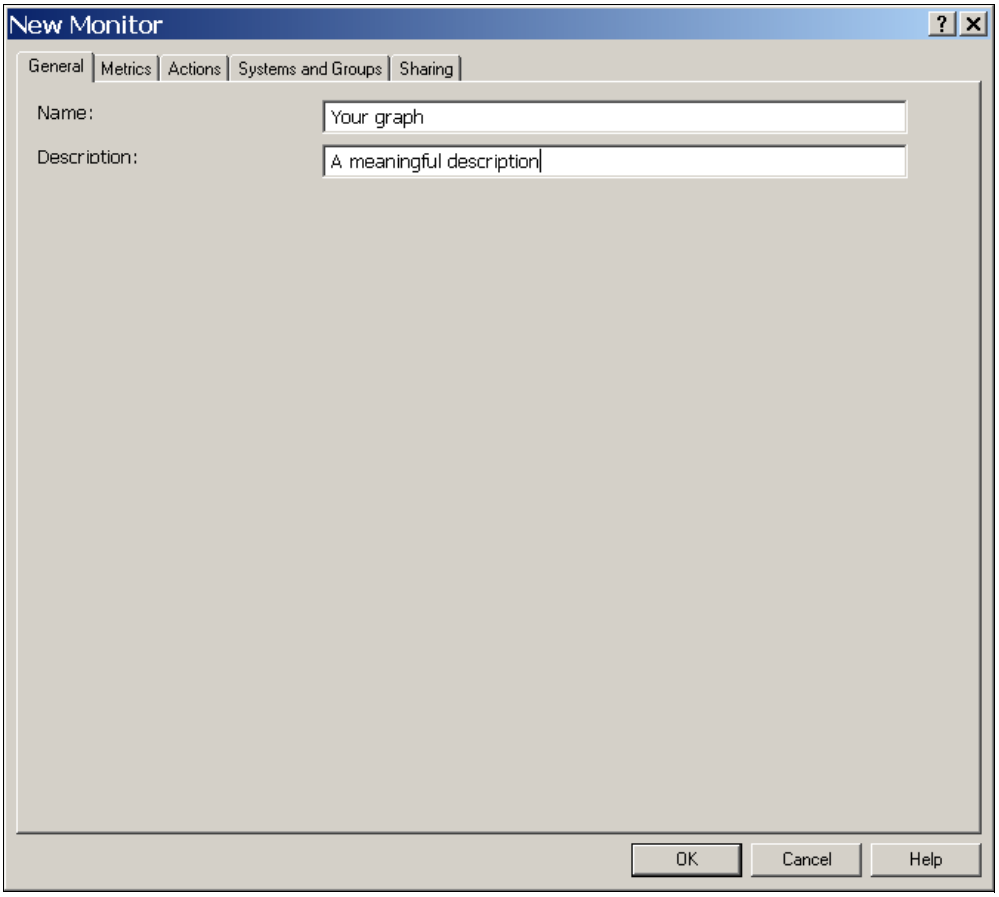


Figure 6-4 Naming the monitor

4. Open the **Metrics** tab, select the metrics, and click **Add** to include the metrics to be monitored. This example includes the average utilization of CPU, disk space, and disk arms. As shown in Figure 6-5, select the metrics first and then click the **Add** button.

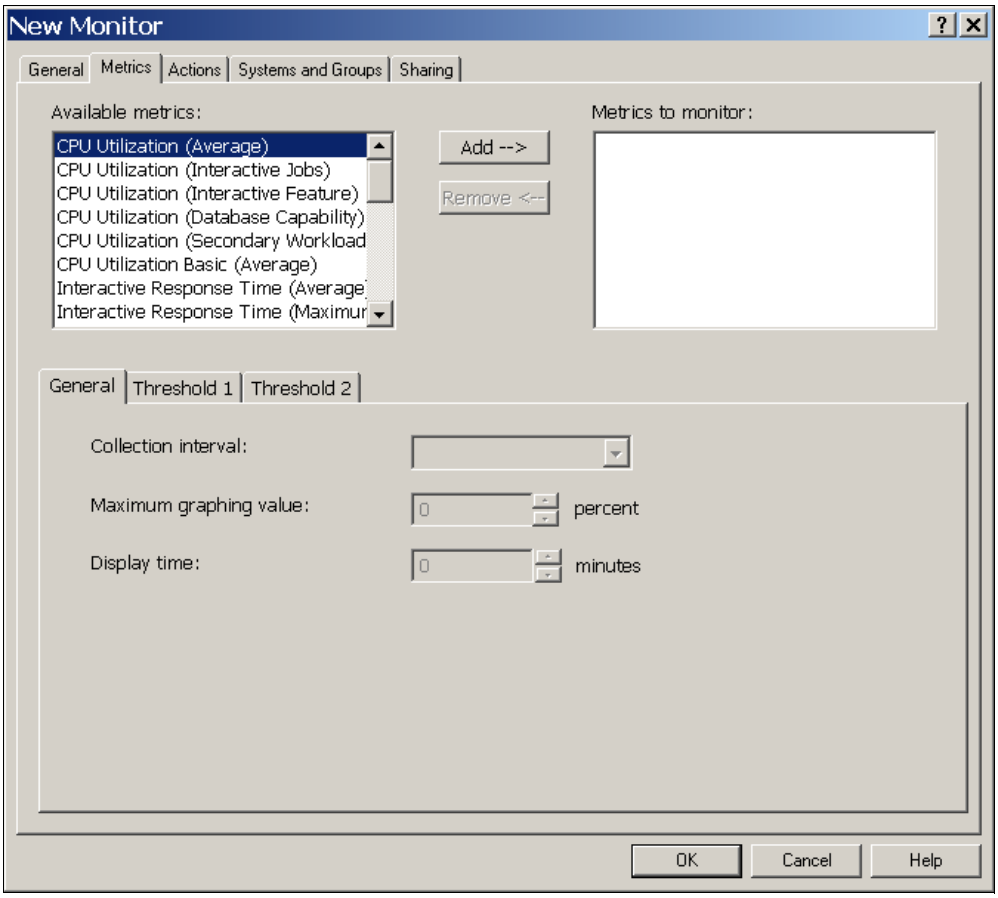


Figure 6-5 Selecting the metrics

5. Select the drop-down menu to select the collection interval, as shown in Figure 6-6.

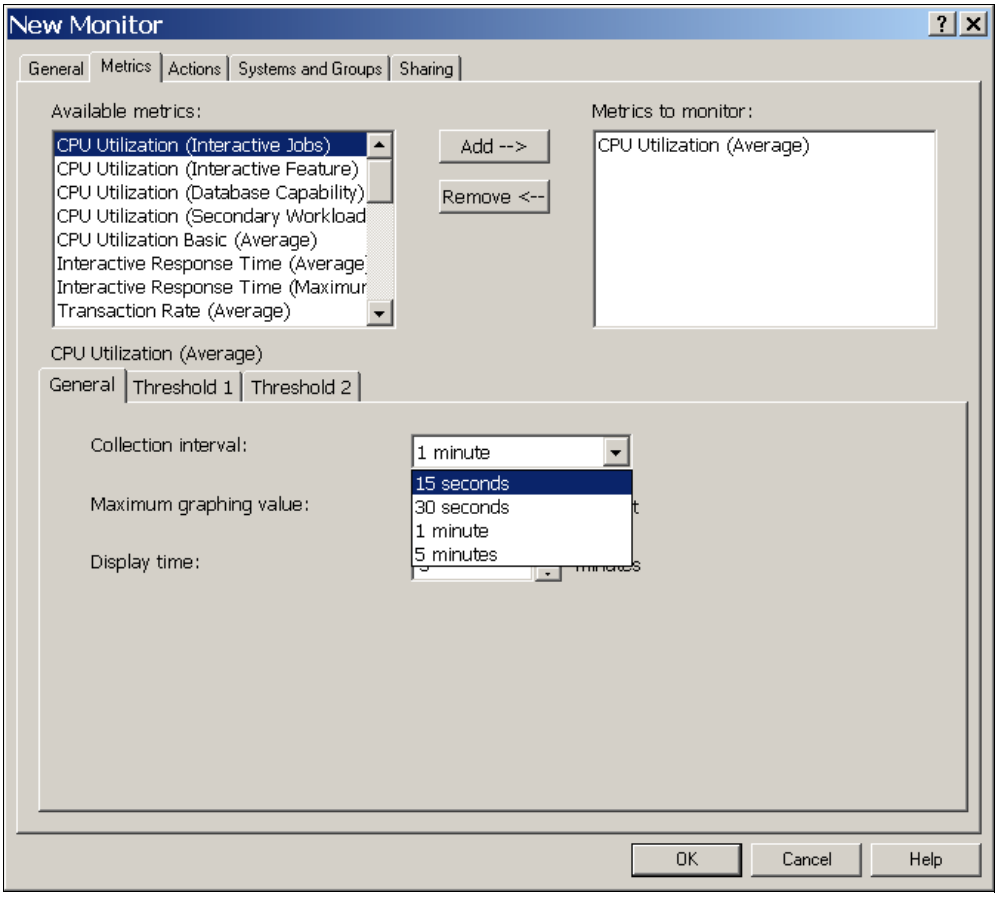


Figure 6-6 Select the collection interval

6. Select the metrics and set the collection interval. Repeat this step until all three metrics are shown in the Metrics to Monitor window, as shown in Figure 6-7. Click **OK**.

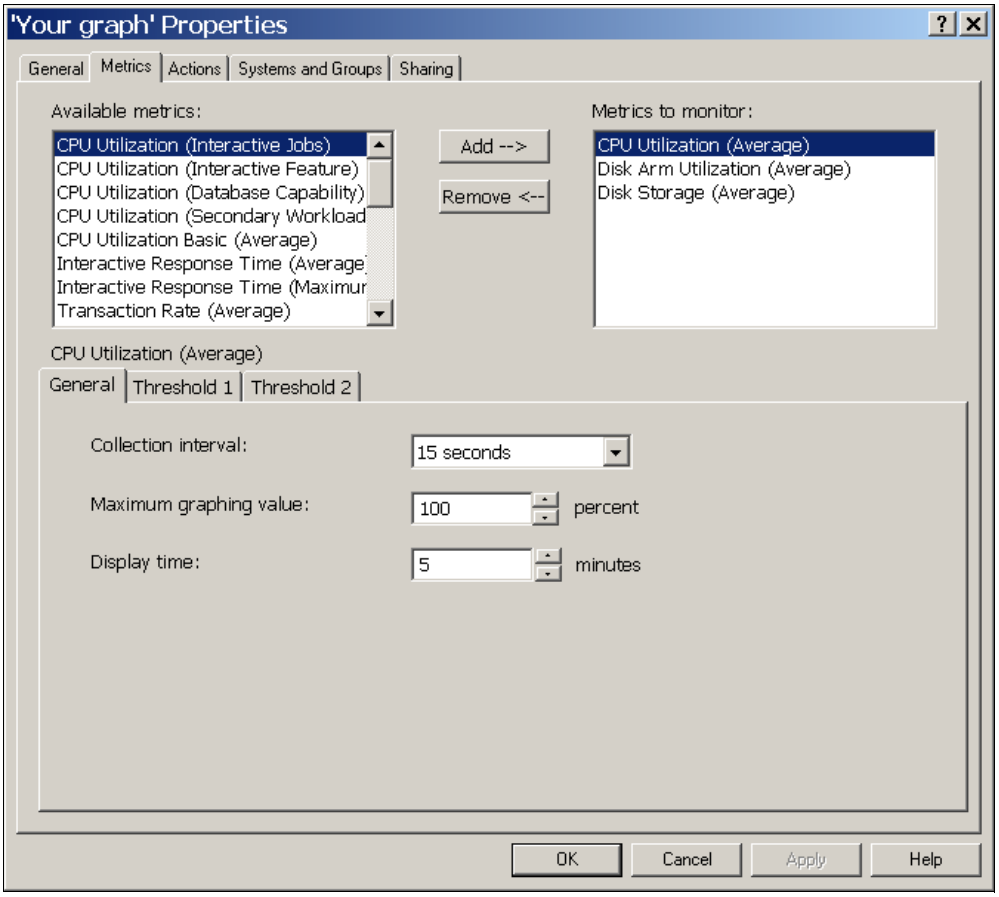


Figure 6-7 The selected metrics

7. After selecting both the metrics and the collection intervals, click the **OK** button to return to the Monitors option that is shown in Figure 6-8.

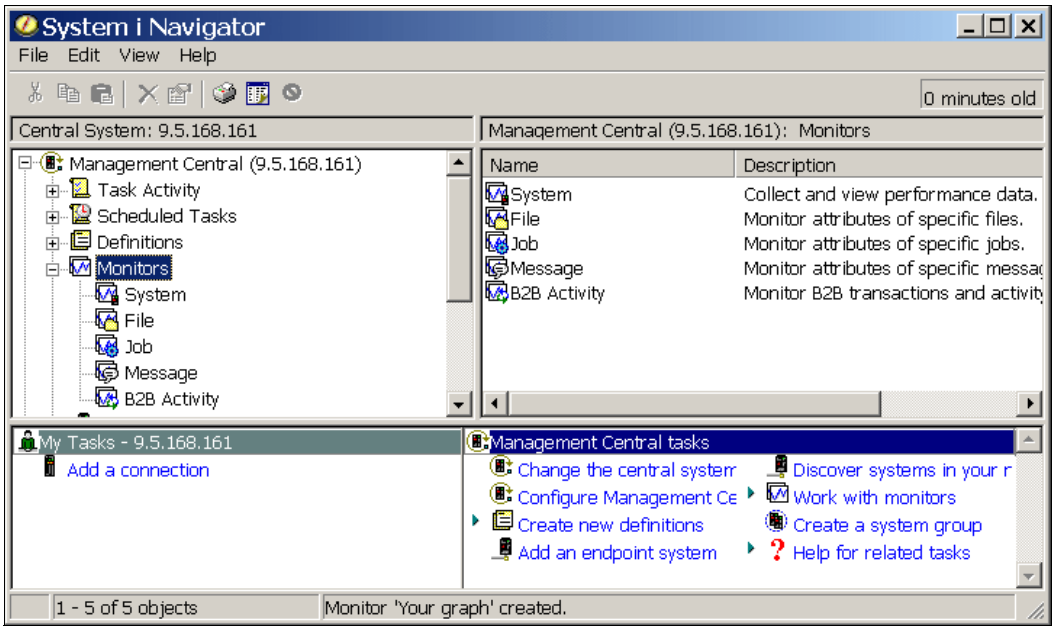


Figure 6-8 Finding the newly created monitor

8. Selecting the System monitors shows the newly created monitor, which has a status of Stopped in the right pane, as shown in Figure 6-9.

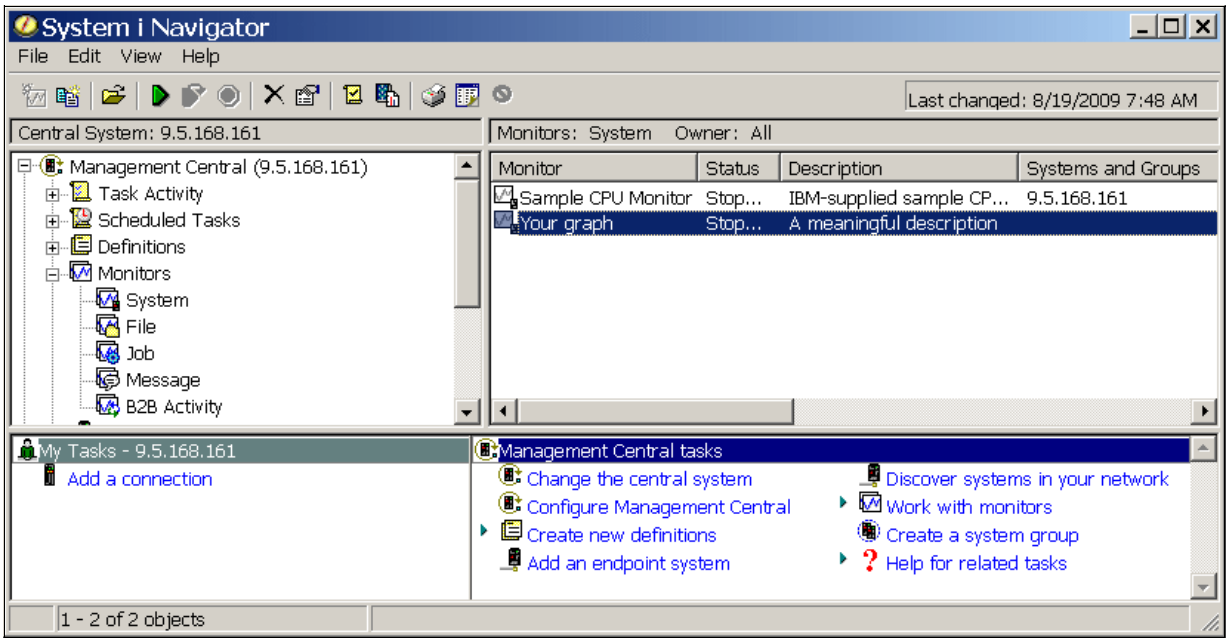


Figure 6-9 Selecting the monitor

9. With the monitor selected, click the green Start button in the upper part of the window to start the monitor, as shown in Figure 6-10.

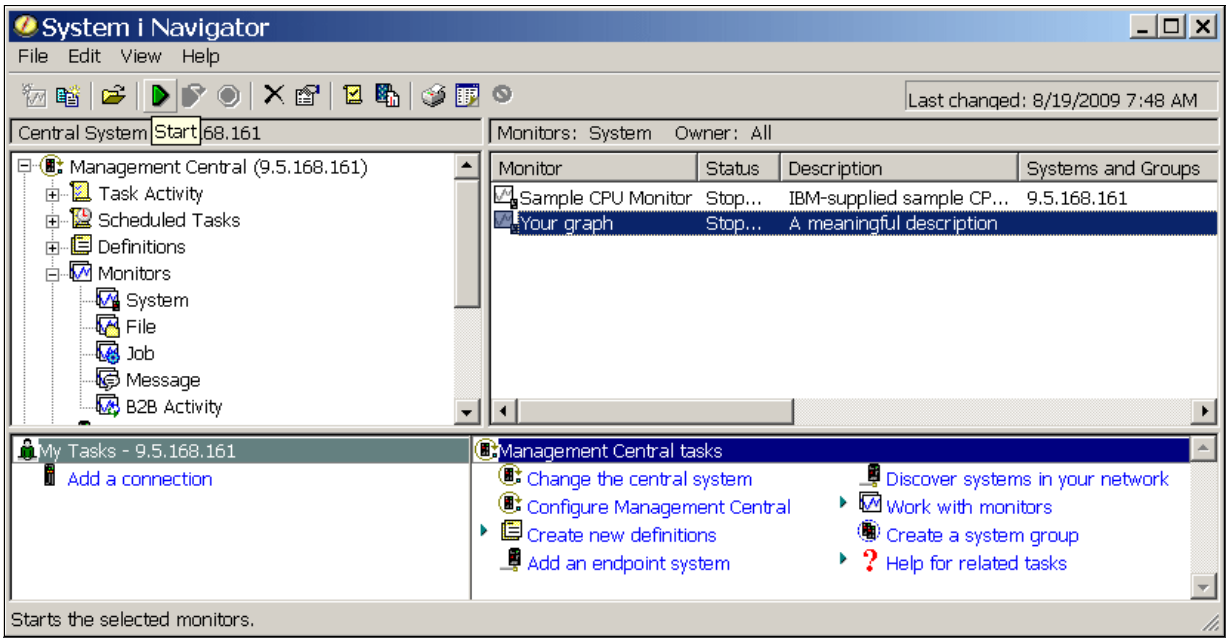


Figure 6-10 Starting the monitor

10. As we did not specify the system when creating the monitor, we are required to do so now. Before starting the monitor, we must specify the system, as shown in Figure 6-11.

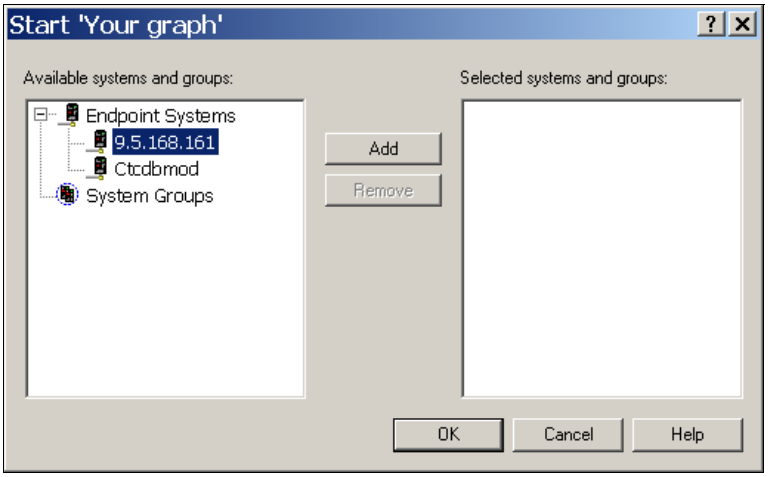


Figure 6-11 Selecting the system

11. Select the system, click **Add**, and click **OK**. The icon in front of the monitor turns green when the monitor is running, as shown in Figure 6-12.

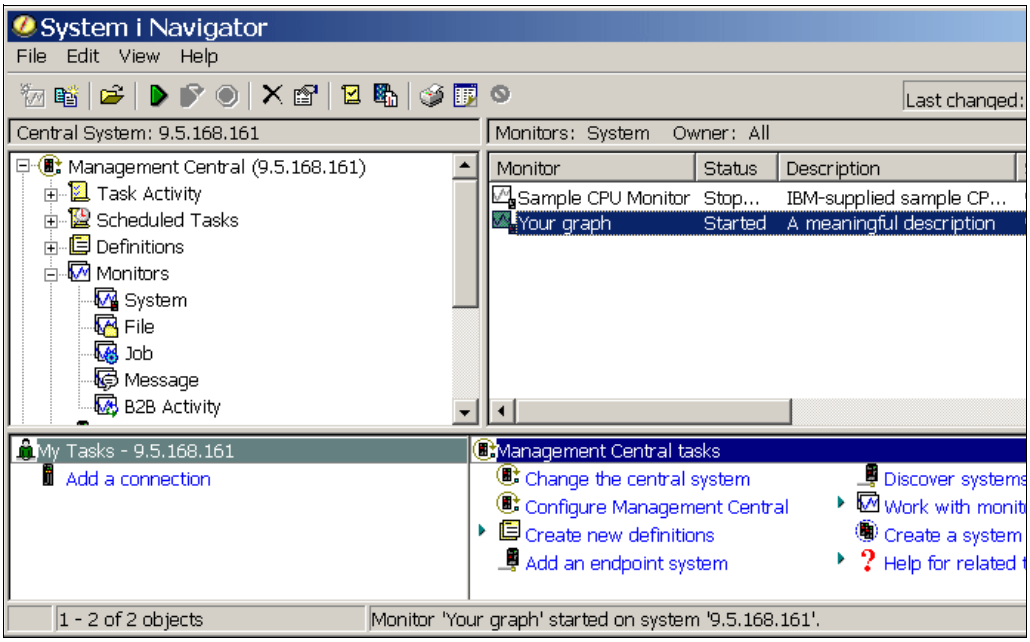


Figure 6-12 Selecting the monitor

The same icon is also shown in the task bar on your workstation. If any of the thresholds are exceeded, the icon turns red.

An example of the monitor that we created is shown in Figure 6-13.

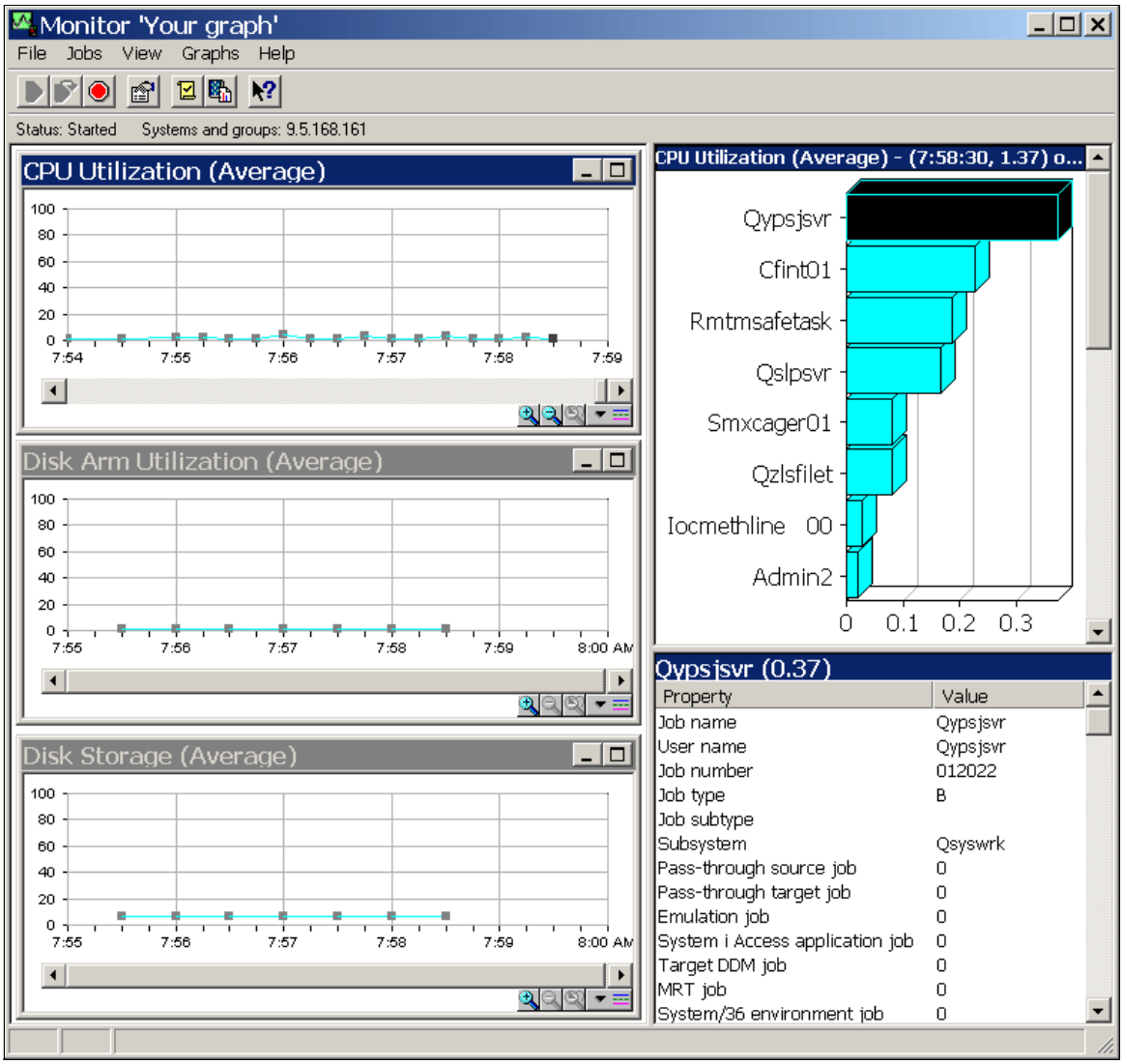


Figure 6-13 A sample monitoring window

Clicking a dot in the CPU graph displays the heaviest jobs using the CPU during that measurement time. Click the last dot to receive an updated list every 15 seconds.

Clicking the job consuming the most of CPU cycles provides you details about the job. Depending on the size of the monitor window, you see different amounts of information.

6.3 CL commands

There are many commands that can be used to observe the partition's performance interactively by using a 5250 session. We explain the most useful of these commands in this section. In addition to the real-time performance monitoring commands, there are several other commands that can be used for retrieving data on IBM i. For more information, refer to the IBM i V6.1 Information Center and select **Systems management** → **Performance** → **Reference information for Performance** → **CL commands for performance**.

Note: All of the real-time performance related CL commands discussed in this section, including WRKSYSACT, are part of the base operating system.

WRKSYSACT was moved from the Performance Tool product into the operating system as of V6.1.

All of these real-time performance related CL commands work similar to Job Watcher and Collection Services in that they are all sampling based. Each refresh computes the delta information about various performance metrics for the ending interval, and the data is collected in a non-intrusive manner.

When you use these commands, you should observe the system's performance during typical levels of activity. For example, statistics gathered when no jobs are running on the system are of little value in assessing system performance.

Here are a few high level steps that you should perform to observe the system's performance using these commands:

1. Enter the WRKSYSSTS, WRKDSKSTS, or WRKACTJOB command.
2. Allow the system to collect data for a minimum of 5 minutes.
3. Press F5 (Refresh) to refresh the display and present the performance data.
4. Tune your system based on the new data.

Press F10 (Restart) to restart the elapsed time counter.

Use both the Work with System Status (WRKSYSSTS) and the Work with Active Jobs (WRKACTJOB) commands when observing the performance of your system. With each observation period, you should examine and evaluate the measures of server performance against the goals you have set.

Some of the typical measures include:

- Interactive throughput and response time, which are available from the WRKACTJOB display.
- Batch throughput. Observe the AuxIO and CPU% values for active batch jobs.
- Spool throughput. Observe the AuxIO and CPU% values for active writers.
- CPU% used for database processing.
- Storage allocated/deallocated and the percentage of net storage that can be used to monitor temporary storage usage.

Each time you make tuning adjustments, you should measure and compare all of your main performance measures. Make the adjustments and evaluate the results for one metric at a time.

The output from all of these commands can also be redirected to a file for later processing. If you have a custom monitoring solution or wish to develop one, you can also use the QShell utility to run these commands with output generated directly on the IFS. For example, you can set up an autostart job that starts immediately after IPL and invokes a CL program that wakes up every 5 minutes or so and issues the following commands:

```
QSH CMD('system wrksyssts > /home/monitor/sysname_wrksyssts.txt')
QSH CMD('system wrkactjob > /home/monitor/sysname_wrkactjob.txt')
```

The CL program can then FTP these text files to a central server where you can process them and use the data to graph the measurements.

6.3.1 Work with Active Jobs command

The Work with Active Jobs (WRKACTJOB) command allows you to work with performance and status information for the active jobs running on the partition. The sequence of jobs can be changed with the Sequence (SEQ) parameter or by using operations on the Work with Active Jobs screen. Other parameters allow the selection of jobs to be shown on the screen. The selection parameters can also be changed by operations on the screen.

Figure 6-14 shows the Work with Active Job screen.

```
                              Work with Active Jobs
                                                                    CTCDBMOD
                                                          09/02/09  10:13:14
CPU %:    46.3     Elapsed time:   00:05:14     Active jobs:   255

Type options, press Enter.
  2=Change   3=Hold   4=End   5=Work with   6=Release   7=Display message
  8=Work with spooled files   13=Disconnect ...
                    Current
Opt  Subsystem/Job  User        Type  CPU %  Function        Status
       CHGUPD05     MARK_ROY    BCH     6.0  PGM-CHAINUP_CW   RUN
       CHGUPD04     MARK_ROY    BCH     6.0  PGM-CHAINUP_CW   RUN
       CHGUPD02     MARK_ROY    BCH     6.0  PGM-CHAINUP_CW   RUN
       CHGUPD01     MARK_ROY    BCH     6.0  PGM-CHAINUP_CW   RUN
       CHGUPD00     MARK_ROY    BCH     6.0  PGM-CHAINUP_CW   RUN
       CHGUPD03     MARK_ROY    BCH     5.9  PGM-CHAINUP_CW   RUN
       QZDASOINIT   NANDOO      PJ      1.3                   TIMA
       QZDASOINIT   NANDOO      PJ       .4                   TIMW
       QZDASOINIT   NANDOO      PJ       .4                   TIMW
                                                                    More...
Parameters or command
===> ____________________________________________________________________
F3=Exit   F5=Refresh       F7=Find      F10=Restart statistics
F11=Display elapsed data   F12=Cancel   F23=More options   F24=More keys
```

Figure 6-14 Work with Active Jobs screen

6.3.2 Work with System Status command

The Work with System Status (WRKSYSSTS) command allows you to work with information about the current status of the partition. You can use this command for near real-time memory monitoring and it also provides a great deal of information, such as:

- Number of jobs currently in the partition.
- Total capacity of the system auxiliary storage pool (ASP).
- Percentage of the system ASP (= disk space) currently in use.
- Amount of temporary storage currently in use.
- Percentage of machine addresses used.

- Statistical information related to each storage pool that currently has main storage allocated to it.

The statistical information is gathered during an identified time interval that is shown as the elapsed time. The data can either be updated by extending the measurement time interval, or the collection can be restarted to show statistics for the interval starting with the previous screen.

The current memory usage is shown for each IBM i storage pool. Pressing F5=Refresh updates the screen with current information. Pressing F10=Restart restarts the statistics from the last displayed time.

Note: When using IBM i automatic performance adjustment, which is enabled by default by the system value setting `QPFRADJ=2`, the memory distribution and activity level for the pools are managed dynamically by the system, depending on the workload.

Figure 6-15 shows the Work with System Status screen.

```
                            Work with System Status
                                                                    CTCDBMOD
                                                          09/02/09  10:30:23
% CPU used . . . . . . . :       43.4    Auxiliary storage:
% DB capability  . . . . :       36.1      System ASP . . . . . . :     1411 G
Elapsed time . . . . . . :   00:11:20      % system ASP used  . . :     9.8320
Jobs in system . . . . . :        776      Total  . . . . . . . . :     1411 G
% perm addresses . . . . :       .007      Current unprotect used :     8723 M
% temp addresses . . . . :       .013      Maximum unprotect  . . :    12340 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   -----DB-----  ---Non-DB---
 Pool   Size (M)  Size (M)  Active  Fault  Pages  Fault  Pages
   1      442.28    222.64   +++++     .0     .0     .3    2.8
   2    10607.91      5.80     260    1.7   10.1    3.0   12.4
   3     1107.67      <.01     304     .0     .0     .0     .0
   4         .25       .00       5     .0     .0     .0     .0

                                                                     Bottom
Command
===>     ____________________________________________________________________
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Restart   F12=Cancel
F19=Extended system status         F24=More keys
```

Figure 6-15 Work with System Status screen

6.3.3 Work with Disk Status command

The Work with Disk Status (WRKDSKSTS) command allows you to display and work with the performance and status information for the disk units on the partition. You can use this command for near real-time monitoring of storage and disk arm performance.

The current disk I/O workload statistics are shown for each IBM i disk unit. Select F5=Refresh to update the display with current information. Press F10=Restart statistics to restart the statistics from the last displayed time. Press F11=Display storage use to see the disk Protection type and status.

> **Note:** Pressing F11 three times can take you to a view where you can see if the performance is “degraded” due to the cache batteries being depleted.

For information about what each *Type* means and various values in the *Status* column, refer to the man page for the Work with Disk Status (WRKDSKSTS) command.

From Work with Disk Status screen, you can also go directly to the Work with System Status (WRKSYSSTS) screen by pressing F16=Work with system status.

Figure 6-16 shows the Work with Disk Status command.

```
                              Work with Disk Status
                                                                      CTCDBMOD
                                                              09/02/09  10:43:09
 Elapsed time:   00:05:00

                 Size    %     I/O   Request   Read  Write   Read  Write    %
 Unit  Type       (M)  Used    Rqs  Size (K)    Rqs   Rqs     (K)   (K)   Busy
    1  4328   123488   9.8  170.7      7.8     9.8  160.9   42.2    5.7      5
    2  4328   123488   9.8  425.5      6.1     7.6  417.9   32.5    5.6     10
    3  4328   123488   9.8  444.6      6.3     7.2  437.3   43.1    5.7     11
    4  4328   123488   9.8  423.4      6.0     6.2  417.1   31.5    5.6     10
    5  4328   123488   9.8  899.2      5.3     6.4  892.7   37.3    5.0     20
    6  4328   123488   9.8  606.8      5.9     5.2  601.6   40.5    5.6     14
    7  4328   123488   9.8  611.4      5.9     7.6  603.8   34.7    5.6     14
    8  4328   123488   9.8  +++++      4.6     6.0  +++++   40.7    4.5     36
    9  4328   105847   9.8  578.7      5.7     3.5  575.2   26.1    5.6     13
   10  4328   105847   9.8  559.3      5.8     4.0  555.2   40.9    5.6     13
   11  4328   105847   9.8  568.8      5.8     4.0  564.7   41.5    5.6     13
   12  4328   105847   9.8  855.7      5.2     3.5  852.2   40.2    5.0     19

                                                                        Bottom
 Command
 ===> ____________________________________________________________________________
 F3=Exit   F5=Refresh   F12=Cancel   F24=More keys
```

Figure 6-16 Work with Disk Status screen

6.3.4 Work with System Activity command

The Work with System Activity (WRKSYSACT) command allows you to interactively work with the jobs and tasks currently running in the system. The Work with System Activity screen shows the current physical and, if configured, virtual processor allocation to the IBM i partition and CPU utilization percentages for average, maximum, and minimum utilizations as well as the CPU utilization by all jobs performing database processing work (Overall DB CPU util). Additionally, a list of jobs or system tasks sorted with the ones consuming the most processing time listed at the top is shown when the default sort of *CPU is used for the *Sequence* parameter.

Pressing F10=Update list refreshes the display with the current information. The refresh interval should be 5 seconds or longer in order for the sampled data to be statistically meaningful. Using the F19=Automatic refresh function for automatically refreshing the screen with the specified interval duration in Automatic refresh in seconds allows you to easily identify jobs consuming significant processing time, because they would appear repeatedly in the top part of the list.

Note: The CPU utilization can actually exceed 100%, depending on how the partition is configured.

The Work with System Activity function uses different views to present all the performance statistics. Pressing F11 shows you the next view in the series and pressing F10 refreshes the current view. F11 shows you these various views:

- View 1: CPU utilization, database utilization, and the number of asynchronous and synchronous I/O operations
- View 2: Synchronous database and non-database read and write operations
- View 3: Asynchronous database and non-database read and write operations
- View 4: Storage allocations (amount of storage that has been assigned or released)

For more information about the Work with System Activity command, refer to Chapter 6, “System Activity”, in *Performance Tools for iSeries*, SC41-5340.

Note: The Work with System Activity command is included in the base operating system on IBM i V6.1. Prior to V6.1, the Performance Tools license program was required. This command is a thread listing command instead of a job one like WRKACTJOB.

Figure 6-17 shows the Work with System Activity screen.

```
                          Work with System Activity
                                                                    CTCDBMOD
                                                            09/02/09  10:41:33
Automatic refresh in seconds  . . . . . . . . . . . . . . . . . . .     5
Job/Task CPU filter . . . . . . . . . . . . . . . . . . . . . . . .    .10
Elapsed time . . . . . . :   00:00:03    Overall CPU util . . . . :    81.5
Overall DB CPU util  . . . :    46.3

Current processing capacity:     1.00

Type options, press Enter.
  1=Monitor job   5=Work with job
                                                          Total  Total   DB
     Job or                                          CPU   Sync  Async   CPU
Opt  Task         User        Number  Thread    Pty   Util   I/O    I/O   Util
     QIDRJWSUM    NANDOO      018604  0000000F    1   34.0    765   1261  31.7
     CHGUPD00     MARK_ROY    018408  0000000E   50    4.6    937   3568    .0
     CHGUPD04     MARK_ROY    018412  00000016   50    4.6    940   3201   1.1
     CHGUPD01     MARK_ROY    018409  00000004   50    4.6    939   3138   3.7
     CHGUPD05     MARK_ROY    018413  00000004   50    4.5    935   2958   2.4
     CHGUPD03     MARK_ROY    018411  00000009   50    4.5    931   3034    .0
                                                                     More...
F3=Exit   F10=Update list   F11=View 2   F12=Cancel   F19=Automatic refresh
F24=More keys
```

Figure 6-17 Work with System Activity screen

6.4 IBM Systems Director Navigator for i

The Systems Director Navigator for i interface and the Systems i Navigator interface are very similar; one is rendered for the Web and the other one is rendered for a Windows client. The features between the two are different. Some things are only resident in the System i Navigator Client (Management Center and DB tools) and some features are only resident in the Systems Director Navigator for i interface (new performance tasks).

There is also the IBM i character based interface. IBM Systems Director Navigator for i provides a graphical interface that is similar but more powerful than the CL commands discussed earlier. The Web-based interface provides additional functionality, such as:

- The ability to look at the last SQL statement executed
- Server information for a job
- Capability to filter and sort the list of items
- Configuring the list to display additional columns
- Easier navigation within the list, including the capability to jump to a particular page

6.4.1 Performance tasks: real-time status

There are three performance tasks available under the Performance tab of IBM Systems Director Navigator for i that monitor real-time status:

1. System Status
2. Disk Status
3. Active Jobs

To bring up the tasks, click **Performance** under the i5/OS Management tree, click **Show All Performance Tasks**, and then select the desired task from the context menu of the Performance menu, as shown in Figure 6-18.

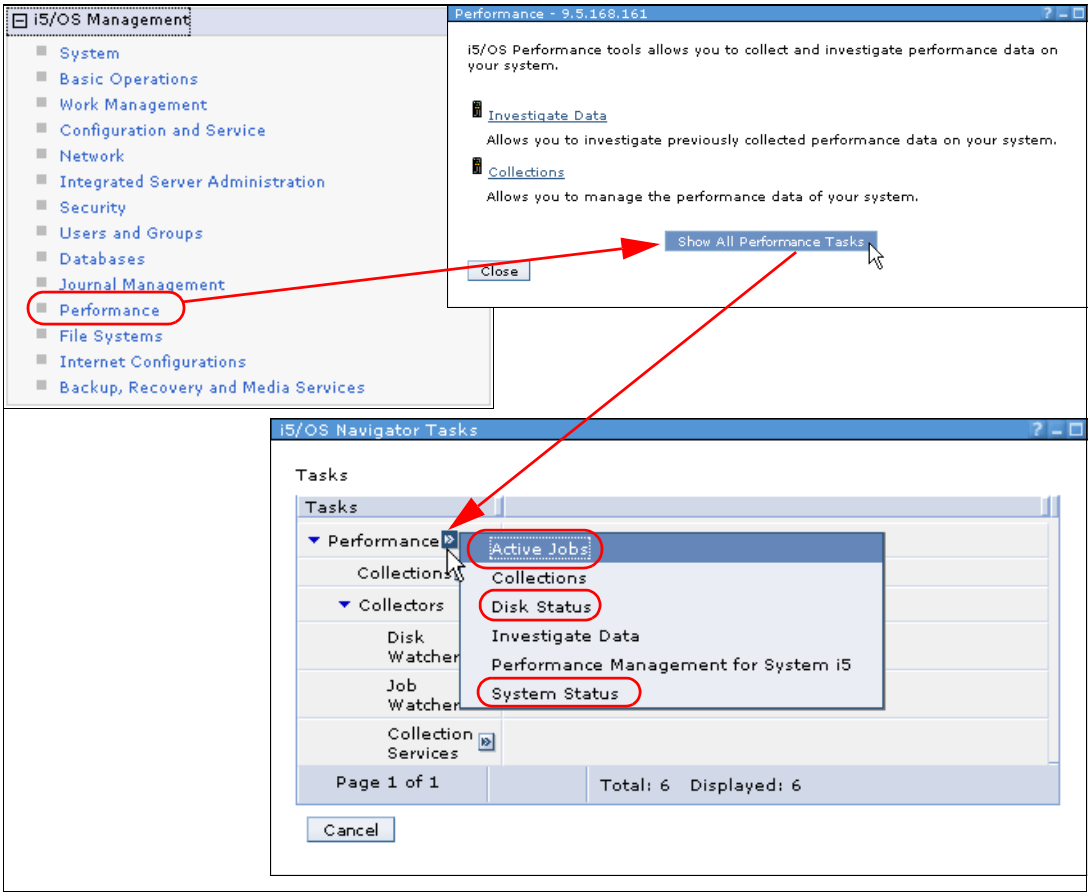


Figure 6-18 Accessing the task for working with real-time performance

The same real-time performance tasks are accessible from the System and Work Management menus under the i5/OS Management tree, as shown in Figure 6-19.

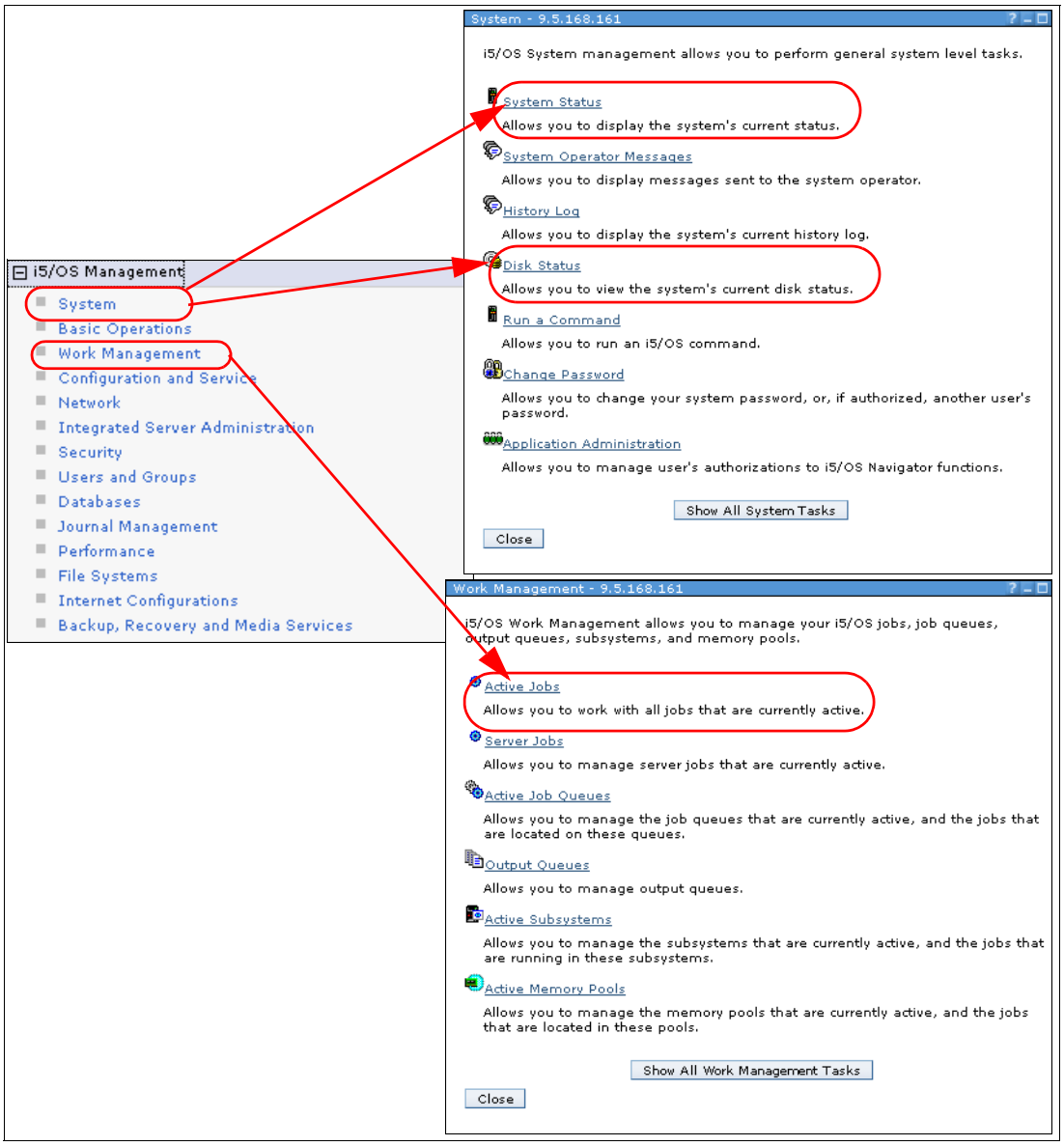


Figure 6-19 Alternate path to get to real-time performance

Active Jobs

The Active Jobs window is similar to the WRKACTJOB command screen. It allows you to work with performance and status information for the active jobs running on the partition. Unlike the WRKACTJOB command screen, the list of active jobs on a graphical interface can be filtered, sorted, and navigated very easily. You can also see the last executed SQL for a particular job in addition to accessing the call stack and performance statistics. You can customize the window to display only the columns you want. For example, you can add a Server column to see which server function to which a job belongs.

A sample Active Jobs window is shown in Figure 6-20

The table window that shows the list of active jobs is similar to other tables throughout IBM Systems Director Navigator for i. For more information about how to customize and navigate the list, refer to 4.6.1, “Working with the Collections interface” on page 70.

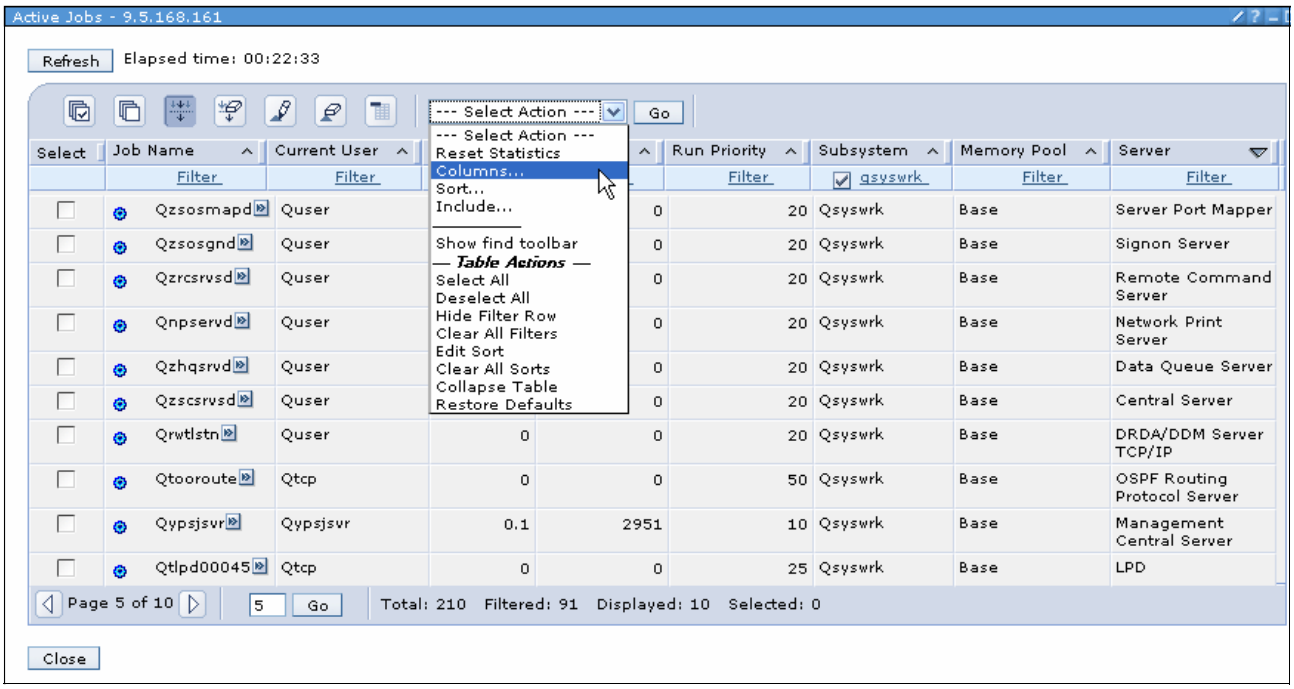


Figure 6-20 Active Jobs window

Disk Status

The Disk Status task is very similar to the Work with Disk Status(WRKDSKSTS) command. It shows usage statistics and I/O rates for each disk unit and the Refresh button can be used to take a sample of the statistics. The Reset Statistics option in the Actions drop-down menu will restart the statistics. On a system with numerous disk units, you can use the filtering capability of the list to monitor a handful of disk units closely. A sample Disk Status window is shown in Figure 6-21.

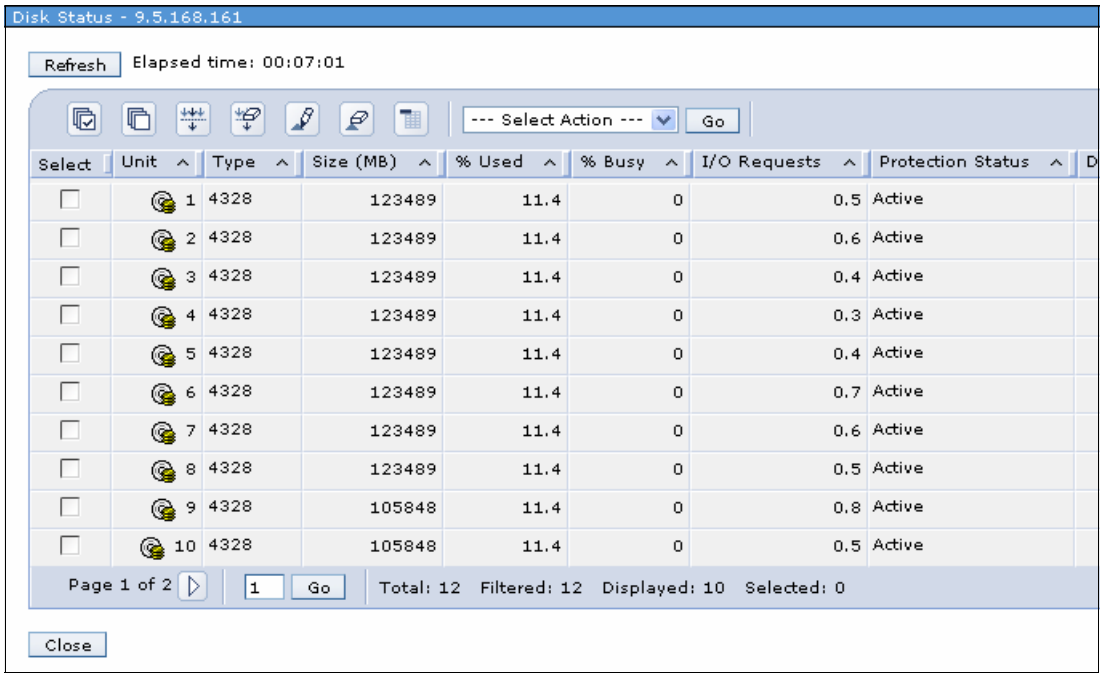


Figure 6-21 Disk Status window

System Status

The System Status task on IBM Systems Director Navigator for i has all the information that is available by running the WRKSYSSTS command, but it is more powerful and easier to use compared to the Work with System Status (WRKSYSSTS) command because all the related options, such as Active Jobs, Job System Values, Active Memory™ Pools, Disk Status, and Storage System Values, are accessible directly from this window. The System Status information can be sampled by clicking the **Refresh** button. The statistics can be restarted by clicking the **Reset Statistics** button at the bottom of the window. Various statistics and options are grouped into category tabs in the System Status window, as shown in Figure 6-22.

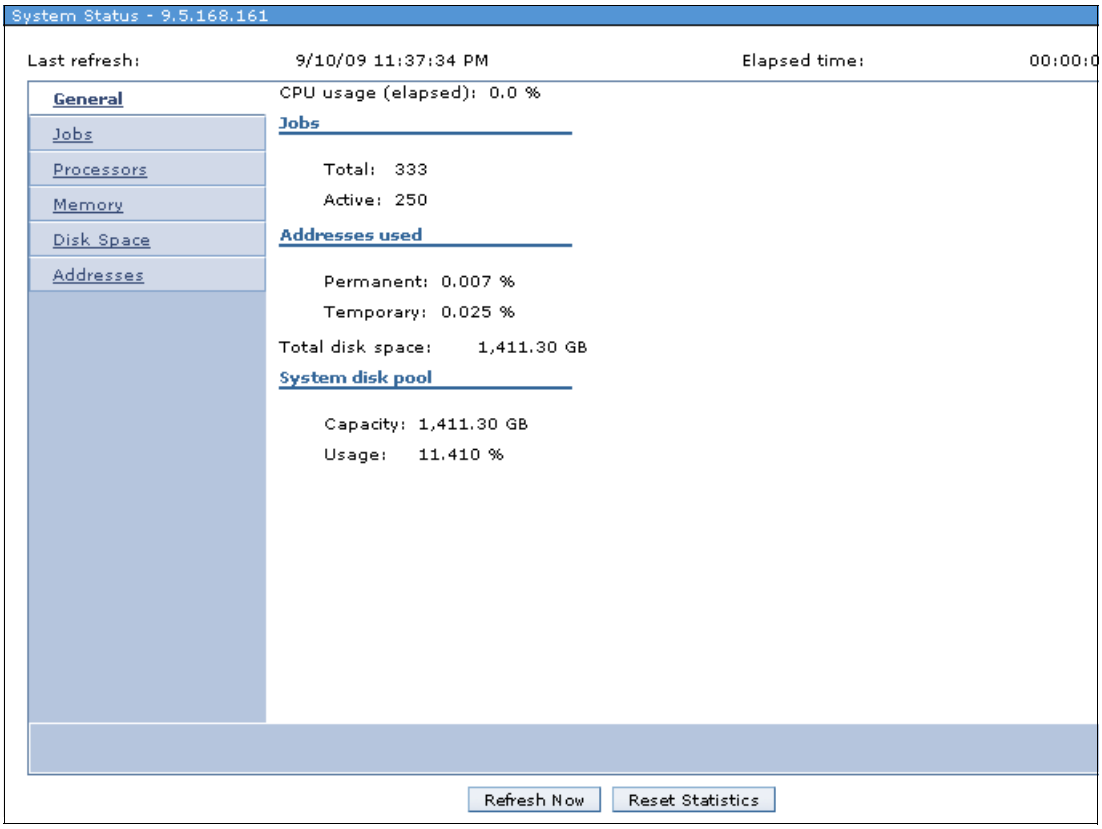


Figure 6-22 System Status window

6.4.2 Health Indicators

The Health Indicators in Performance Data Investigator gives you a graphical view of the general health of the system. By default, it uses the most recent Collection Services collection for the charts, but you can view the Health Indicators for any Collection Services performance collection (*CSFILE) that exists on the system.

The Health Indicators content package has five perspectives:

- System Resources (Default Perspective)
- CPU
- Disk
- Memory
- Response Time

You can access the Health Indicators within Performance Data Investigator by following the steps shown in Figure 6-23.

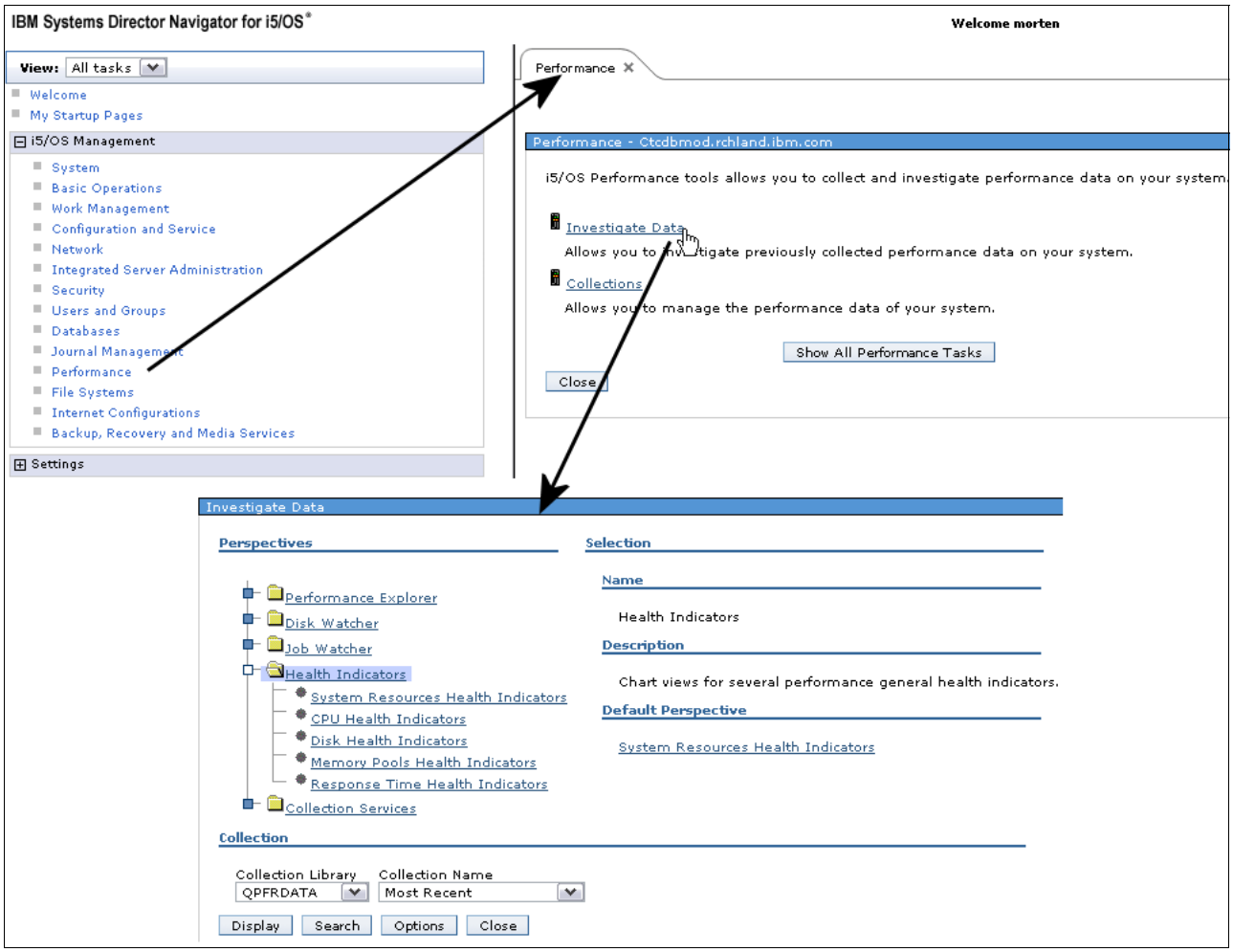


Figure 6-23 Health Indicators

Working with Health Indicators

Health Indicators perspectives are a good starting point when you want to analyze or monitor the performance of your system.

The perspectives within Health Indicators graphically indicate if the resources are under or above the guidelines and thresholds. The percentage of intervals with values under the defined thresholds is shown in green, the percentage of intervals with values above the Warning threshold is shown in yellow, and the percentage of intervals with values above the Action threshold are shown in red.

The guidelines for various resources are predefined, but you can change the guidelines based on any Service Level Agreements (SLAs) you may have in your organization.

Defining Health Indicators

The threshold values for Warning and Action levels for various performance metrics within Health Indicators can be defined from any of the Health Indicator perspectives. However, the Define Health Indicators option in the Actions drop-down menu on the System Resources Health Indicators page, shown on Figure 6-25 on page 157, provides the ability to define thresholds for all Health Indicators perspectives. They are all grouped and presented in multiple tabs, as shown in Figure 6-24.

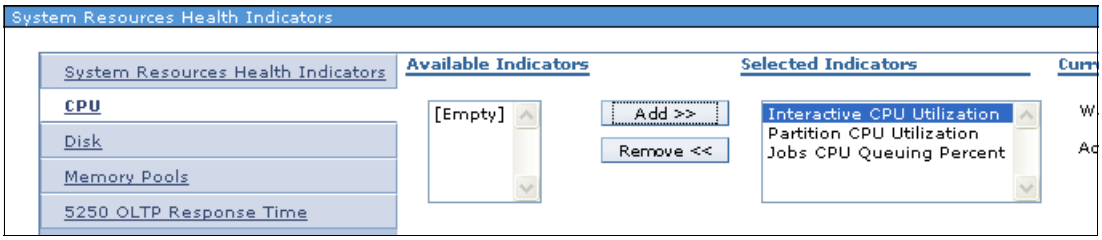


Figure 6-24 Define Health Indicators for system resources

System Resources Health Indicators

The System Resources Health Indicators perspective gives you an excellent overview of your Health Indicators by importing information from the other perspectives of the Health Indicators package. You can determine the general health of main system resources according to the defined thresholds using this perspective.

From the System Resources Health Indicators perspective, you can drill down into any of the other perspectives and further drill down into the Collection Services perspectives for detailed analysis.

A sample System Resources Health Indicators window is shown in Figure 6-25.

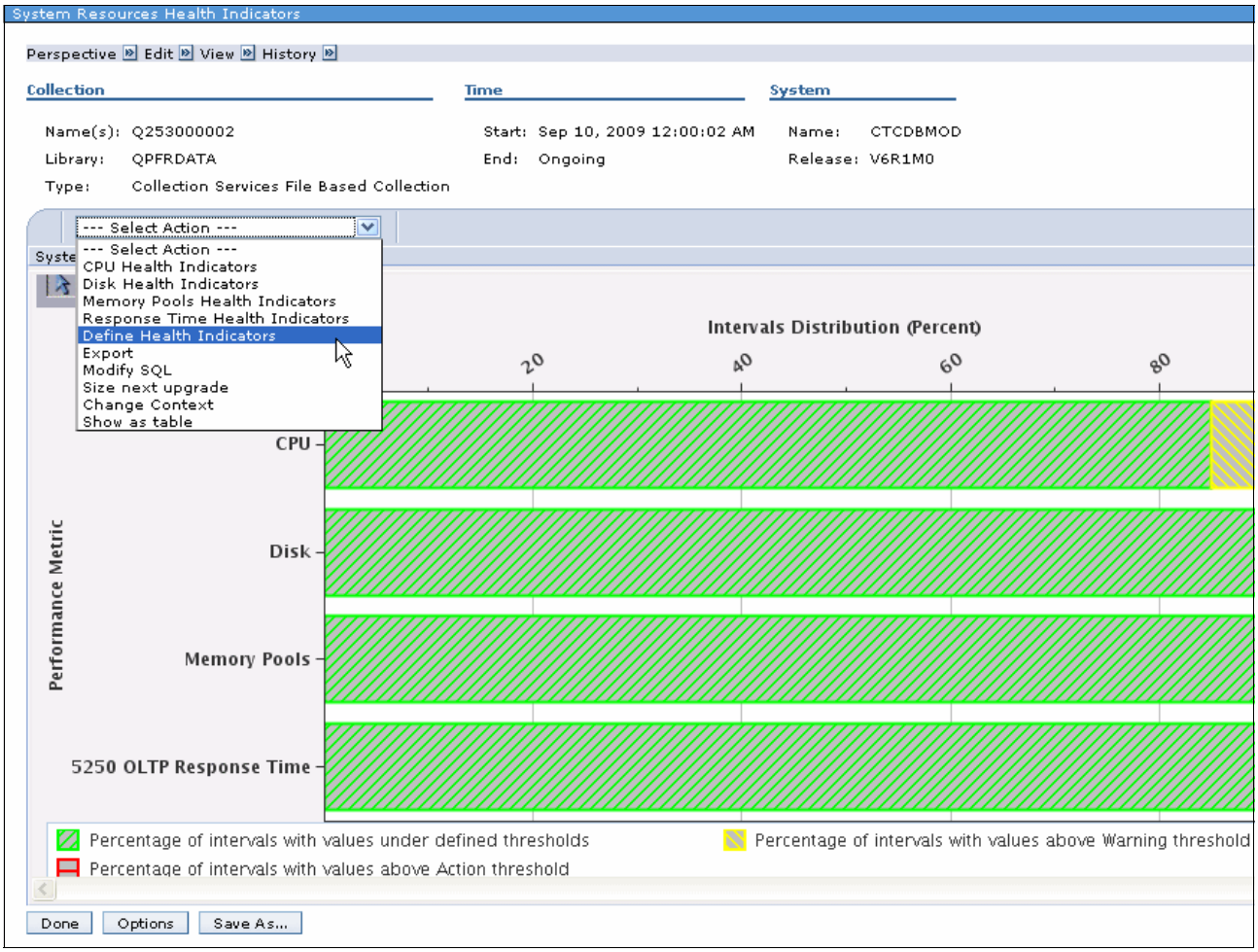


Figure 6-25 System Resources Health Indicators

For more information about how you can use Health Indicators to investigate a performance problem, refer to 5.3.1, “Investigate using Health Indicators and Collection Services” on page 92.

CPU Health Indicators

The CPU Health Indicators perspective shows three indicators:

- Partition CPU Utilization
- Jobs CPU Queuing Percent
- Interactive CPU Utilization

Disk Health Indicators

The Disk Health Indicators perspective shows three indicators:

- Average Disk Space Percent Used
- Average Disk Space Percent Busy
- Average Disk Response Time

Memory Pools Health Indicators

The Memory Pools Health Indicators perspective shows two indicators:

- Page Faults Pending Per Second
- Page Faults Per Second

Response Time Health Indicators

This perspective shows 5250 OLTP Response Time.

6.5 Common Information Model

The Common Information Model (CIM), adopted and evolved by the Distributed Management Task Force (DMTF), is a published systems management standard and was developed in an open forum by DMTF member companies. Defined and promoted as an industry standard for managing systems, CIM was designed to be used for describing management information between differing management applications, running in many different operating environments, including Microsoft® Windows and Linux®. Detailed information about the CIM Specification is available from the DMTF home page at the following address:

http://www.dmtf.org

CIM provides a common definition of management information for systems, networks, applications, and services, and allows for vendor extensions. CIM's common definitions enable vendors to exchange rich management information between systems throughout the network. CIM is composed of a specification and a schema. The schema provides the actual model descriptions, while the specification defines the details for integration with other management models.

CIM is used extensively throughout Systems Director. On IBM i V6.1, CIM is a feature of the IBM Universal Manageability Enablement licensed program (5722-UME). It includes the following functions:

- Provider instrumentation for server resources on the system. The providers, which are based on a subset of the standardized CIM classes, gather data on a system.
- Common Information Model Object Manager (CIMOM), a central component of the WBEM server that is responsible for the communication between clients and providers. CIMOM also provides several management functions, including security, and a set of commands that provide configuration and management functions to administrators.
- A schema that defines an information model for representing systems management functions.
- An implementation of the standardized formats for communication between clients and CIMOM, called CIM in XML, V2.1 and CIM operations over HTTP, V1.1. For more information about these standards, see the WBEM Web site at the following address:

  http://www.dmtf.org/standards/wbem/

6.6 IBM Systems Director

IBM Systems Director enables monitoring and event management across heterogeneous IT environments from a single browser-based user interface. IBM Systems Director provides broad platform coverage, including Microsoft Windows, Intel® Linux, Power Linux, AIX®,

i5/OS, IBM i, and IBM System z® Linux environments across System p, System i, System x, System z, IBM BladeCenter®, and OpenPower®, as well as x86-based non-IBM hardware.

For IBM i, Platform Agent is part of the Universal Manageability Enablement (UME) in the base operating system. IBM Systems Director provides access to *all* of the Performance tasks available in IBM Systems Director Navigator for i, including Performance Data Investigator and Collection Management.

6.6.1 IBM Systems Director components

IBM Systems Director is designed to manage a complex environment that contains numerous servers, desktop computers, workstations, notebook computers, storage subsystems, and various types of SNMP-based devices. Figure 6-26 shows a simple diagram of the major components that you might find in a Systems Director managed environment, as well as the Systems Director software components that might be installed on each type of hardware.

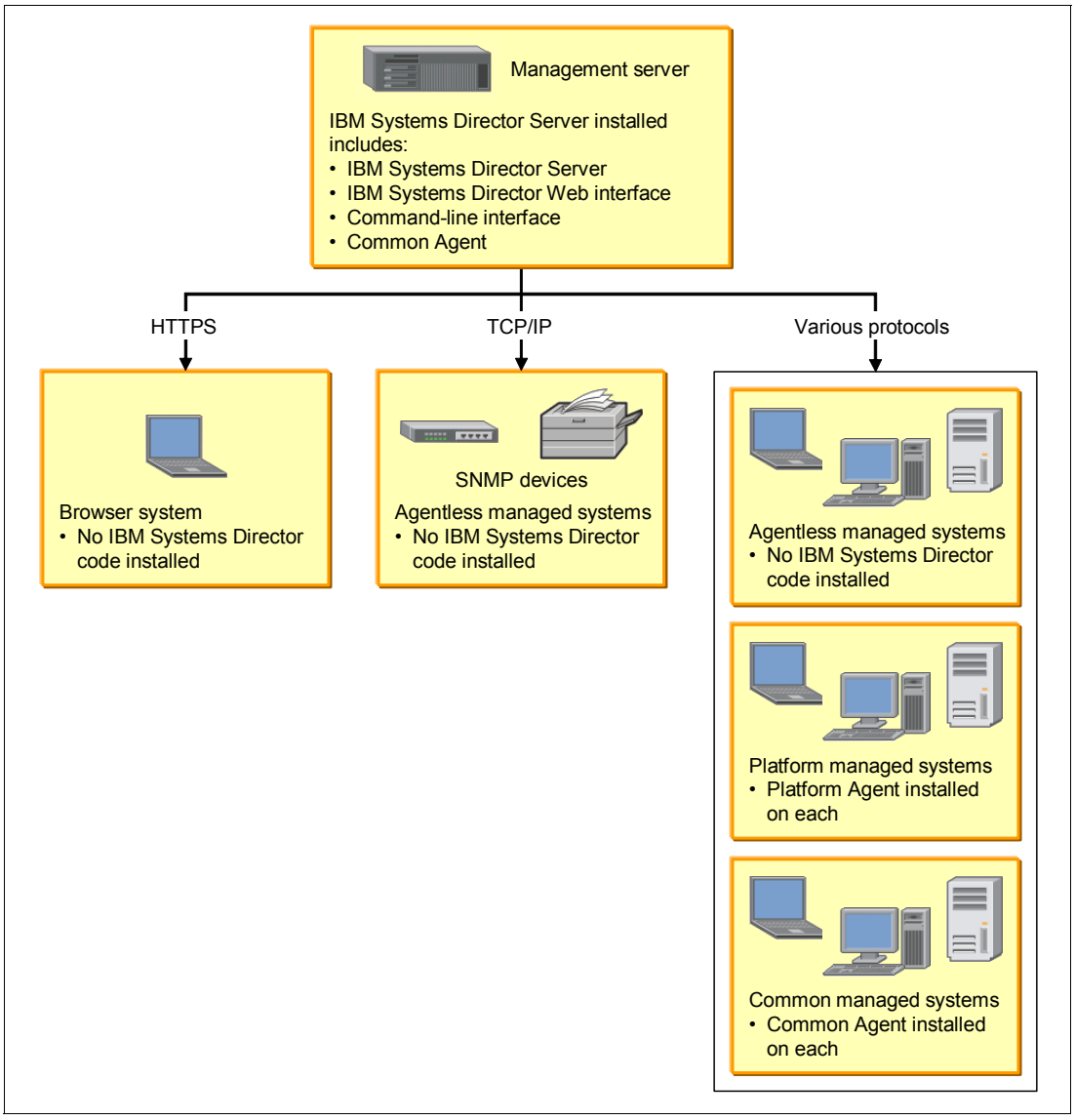


Figure 6-26 IBM Systems Director management environment showing components

Note: IBM Systems Director V6.1 Server is not supported on IBM i. Also, there is no Common Agent for IBM i, but Platform Agent could be used instead.

6.6.2 Managing Power Systems

Management of Power Systems servers is simplified through the exceptional value of IBM Systems Director. The tool manages the underlying hardware, operating systems, and platform managers, such as Hardware Management Console (HMC) and Integrated Virtualization Manager (IVM). Key features that enable life cycle management include discovery, health and status monitoring, configuration, updates, and virtualization. It also provides a Welcome Page summary view that is specific to Power Systems servers and functions that are specific to the Power platform.

You can use the Power Systems features within Systems Director to do the following tasks:

- Manage Power environments that include POWER5™ and POWER6® processor-based servers running the AIX, IBM i, or Linux operating systems and the following:
  - Power Systems servers managed by the Hardware Management Console (HMC)
  - Power Systems servers managed by the Integrated Virtualization Manager (including POWER® processor-based BladeCenter servers)
  - A Power Systems server with a single image
  - A POWER processor-based BladeCenter server under the control of a BladeCenter management module
- Perform management tasks on systems that are under the control of the HMC and IVM, including managing power, creating virtual servers, editing virtual server resources, and relocating virtual servers between host systems.
- Visualize relationships between virtual resources and physical resources.
- Perform operating system specific management tasks that are available from the IBM Systems Director Console for AIX and the IBM Systems Director Navigator for IBM i.
- Collect inventory of virtual storage and associated storage pools hosted by IBM i and used by client Power logical partitions (LPARs) and iSCSI-attached Intel architecture BladeCenter blade and System x servers, providing intuitive viewing and management of those hosted servers.

Systems Director V6.1 simplifies IT management for Power Systems users in several key areas:

- Consolidated management of physical and virtual resources
- Health monitoring
- Automation
- Updates
- Energy management

Logging into the Web interface

To log into the IBM Systems Director Web interface, you must open a Web browser and perform these steps:

1. Open the browser and point the browser to the following URL:

   `http://system_name:port_number/ibm/console`

   - *system_name* is the host name (or IP address) of the system on which IBM Systems Director Server is installed.

– *port_number* is the first (lower) of two consecutive port numbers that you configured the Web server to use during installation. (Default ports for the Web server are 8421 for http and 8422 for https.)

2. On the Web console, enter the user ID and password corresponding to an authorized IBM Systems Director administrator user ID and password and click **Log in**.
3. The first time that you log in you will be presented with the welcome window shown in Figure 6-27. The IBM Systems Director Welcome page is used to complete first-time setup steps to make sure that IBM Systems Director and its plug-ins are set up and configured to manage your environment.

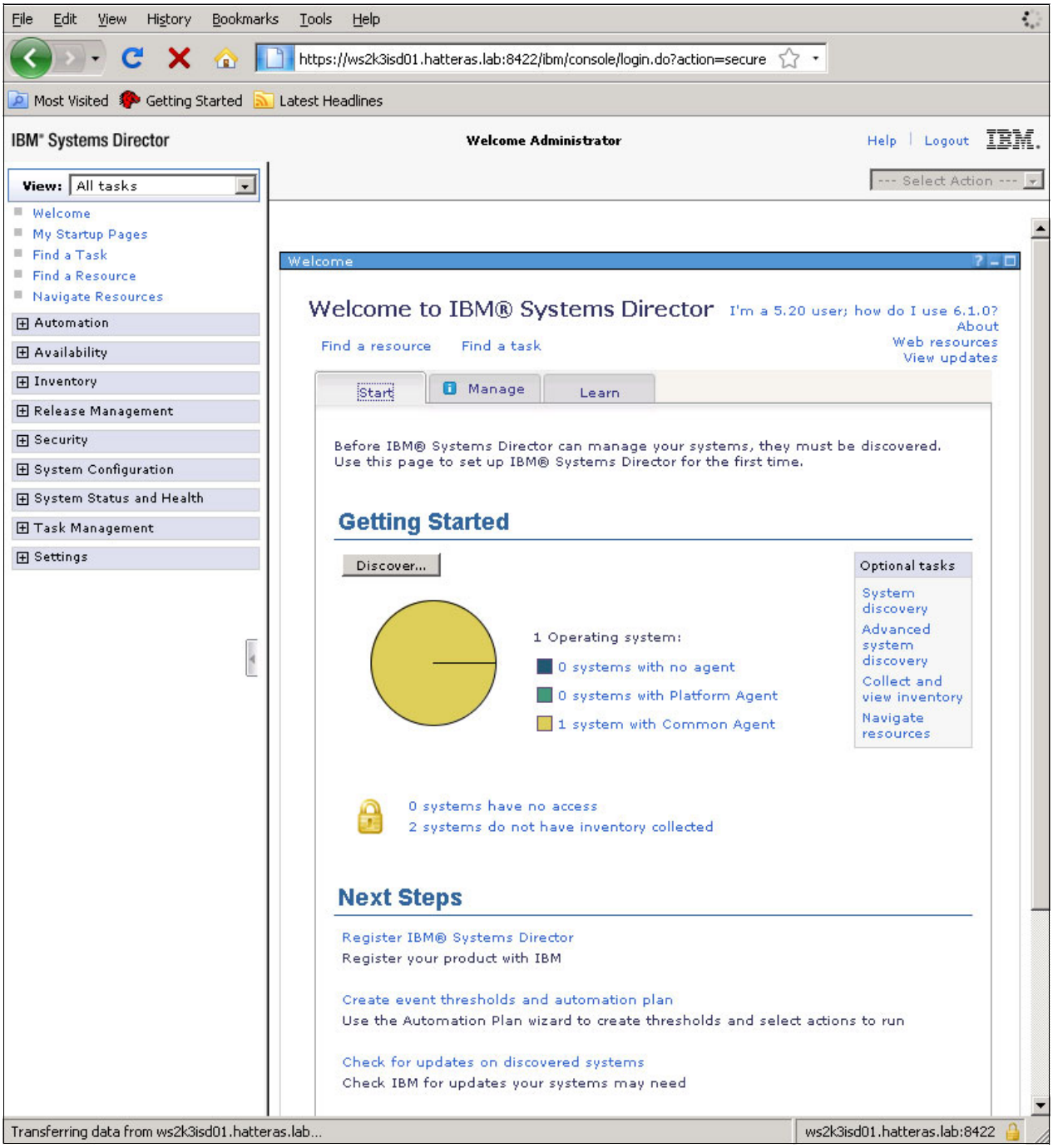


Figure 6-27 Welcome page

4. After the initial discovery is completed, the Manage tab is displayed by default when you view the Welcome page. The Manage tab provides information that you can use to determine whether IBM Systems Director and its plug-ins are installed correctly and ready to use.
5. Click **Power Systems Management**, as shown in Figure 6-28.

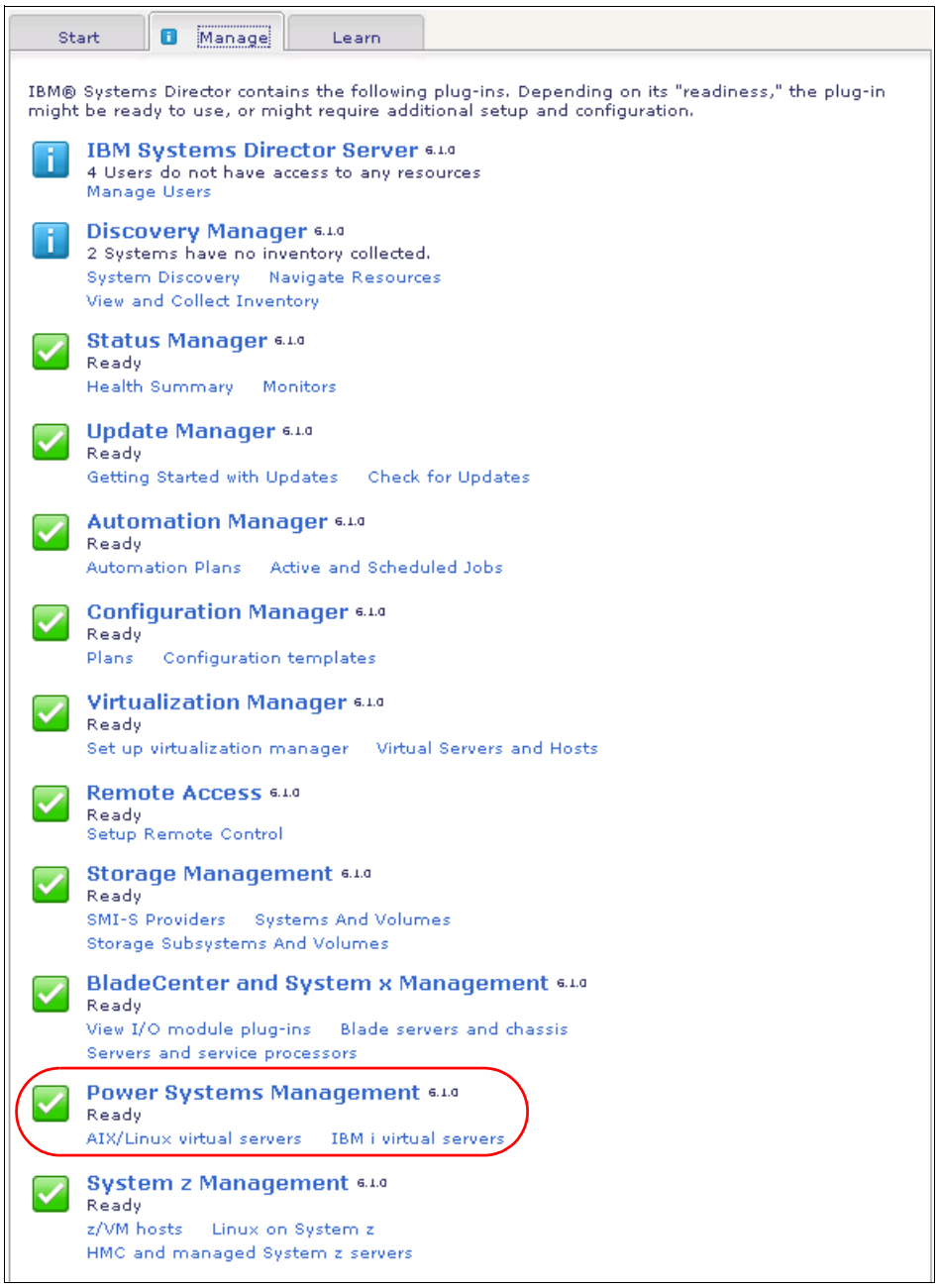


Figure 6-28 Manage tab view

Managing Power Systems running IBM i

The following IBM i systems can be managed by IBM Systems Director V6.1:

- IBM i V6.1 (formerly i5/OS, Version 6 Release 1)
- IBM i V5.4 (formerly i5/OS, Version 5 Release 4)

These IBM i systems can be managed by IBM Systems Director V6.1 using the following applications:

- IBM Director Agent V5.20 for i5/OS
- i5/OS Universal Manageability Enablement 5722-UME, which acts as a Platform Agent
- Agentless support (only discovery, remote session, and a limited subset of the Agent Installation Wizard task)

There is, however, no IBM Systems Director V6.1 Common Agent for IBM i.

IBM Director Agent V5.20 for i5/OS, when managed by IBM Systems Director V6.1, has the following restrictions:

- The Agent Installation Wizard cannot be used to install subagents.
- Configuration Manager: Only operating system configuration plug-ins are supported.
- Update Manager: Only IBM i operating system updates are supported.

For detailed instructions about installing and configuring each component of IBM Systems Director on systems running IBM i, refer to the *IBM Systems Director for IBM i Planning, Installation, and Configuration Guide*, found at the following address:

http://publib.boulder.ibm.com/infocenter/systems/topic/director_6.1/fqp0_bk_install_gde_ibmi.pdf

IBM i management tasks

IBM Systems Director includes IBM i management capabilities for managing IBM i V5.4 or V6.1 endpoint systems using the IBM Systems Director Navigator for i. IBM i users who have not yet moved to the IBM i V6.1 release can now manage V5.4 systems from the Web by using IBM Systems Director V6.1. To access IBM i management tasks in IBM Systems Director, select the IBM i V5.4 or V6.1 system you want to manage. The IBM i Management functions shown in Table 6-1 are enabled for your use.

Table 6-1 IBM i Management Functions available in IBM Systems Director

IBM i Management Task	Description
System	These commonly used System tasks include viewing system status, messages, disk status, and changing passwords.
Basic Operations	These are basic tasks that easily manage lists of messages, printer output, and printers.
Work Management	This function controls the work performed on the system, including support for interactive and batch work. Distribute resources to ensure applications run well.
Configuration and Service	These are functions to help you manage both hardware and software on your system, including system values, time management, and disk units.
Network	This function controls TCP/IP configuration, remote access services, networking servers, and IP policies.
Integrated Server Administration	This function manages servers hosted by IBM i and enrolls and manages users and groups.
Security	These functions allow you to plan and implement security on your system, authorization lists, object permissions, and cryptographic functions.

IBM i Management Task	Description
Users and Groups	This function displays and manages a complete list of the users and groups, and create, delete, and modify them.
Databases	This graphical interface performs many common database operations, accesses and modifies objects, works with performance monitors, as well as accesses the Health center.
Journal Management	This function works with journals and journal receivers.
Performance	This function collects performance data and views the collection to investigate potential performance issues.
File Systems	This function provides links to the most commonly used File Systems tasks, integrated file system, IBM i NetServer, Network File System.
Internet Configurations	These are Internet configuration tasks, such as HTTP Server, digital certificates, IBM IPP, Web-based help server, and Internet Setup wizard.

Upward integration

IBM Systems Director lets you leverage your existing enterprise and workgroup management software by upwardly integrating with those products. IBM Systems Director Upward Integration Modules (UIMs) and management packs allow the supported management systems to interpret and display data provided by Common Agents and Platform Agents. IBM Systems Director UIMs and management packs allow the related products to collect inventory, view IBM Systems Director events, and, for some UIMs, distribute IBM Systems Director software packages.

When planning your IBM Systems Director installation, determine what features you already have implemented in another system, whether that system is supported by an IBM Systems Director UIM or management pack, and whether you want information from IBM Systems Director to flow up to that software. For more information, refer to the IBM Systems Director UIMs section of the IBM Systems Director V6.1 Information Center, which can be found at the following address:

http://publib.boulder.ibm.com/infocenter/systems/topic/director_6.1/fqm0_main.html

6.7 IBM Tivoli Monitoring

IBM Tivoli Monitoring provides a way to monitor the availability and performance of all the systems in your enterprise from one or several designated workstations. It also provides useful historical data that you can use to track trends and to troubleshoot system problems. You can use IBM Tivoli Monitoring to do the following tasks:

- Monitor for alerts on the systems that you are managing by using predefined situations or custom situations.
- Establish your own performance thresholds.
- Trace the causes leading to an alert.
- Gather comprehensive data about system conditions.
- Use policies to perform actions, schedule work, and automate manual tasks.

The Tivoli Enterprise Portal is the interface for IBM Tivoli Monitoring products. By providing a consolidated view of your environment, the Tivoli Enterprise Portal permits you to monitor and resolve performance issues throughout the enterprise.

For product information about IBM Tivoli Monitoring, go to:

http://www.ibm.com/software/tivoli/products/monitor

Also, Tivoli Monitoring Information Center can be accessed at the following address:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp?toc=/com.ibm.itm.doc/toc.xml

6.7.1 Monitoring Agent for IBM i

The Monitoring Agent for IBM i provides you with the capability to monitor IBM i resources, and to perform basic actions with IBM i. IBM Tivoli Monitoring is the base software for the Monitoring Agent for IBM i.

The Monitoring Agent for IBM i offers a central point of management for IBM i systems. It provides a comprehensive means for gathering exactly the information you need to detect problems early and prevent them. Information is standardized across all distributed systems so you can monitor and manage hundreds of servers from a single workstation.

Use the Monitoring Agent for IBM i to easily collect and analyze IBM i -specific information, such as:

- Operating system and CPU performance
- IBM i disk information and performance analysis
- Network performance and information, such as topology and status
- Virtual and physical memory statistics
- Disk and database capacity
- Paging information and swap statistics
- Historical data collection for trend analysis and capacity planning

Note: The Monitoring Agent for IBM i comes as licensed program *5724C04* packaged in the IBM Tivoli Monitoring Agents DVD under product code *A4* and it must be copied into a save file (SAVF) before running the RSTLICPGM command.

Benefits of IBM i Agent

The Monitoring Agent for IBM i provides the following benefits:

- Simplifies application and system management by managing applications, platforms, and resources across your system.
- Increases profits by providing you with real-time access to reliable, up-to-the-minute data that allows you to make faster, better informed operating decisions.
- Enhances system performance because you can integrate, monitor, and manage your environment, networks, console, and mission-critical applications. The Monitoring Agent for IBM i alerts the Tivoli Enterprise Portal when conditions in your environment meet threshold-based conditions. These alerts notify your system administrator to limit and control system traffic. You can view data gathered in reports and charts, informing you of the status of managed resources.

- Enhances efficiency by monitoring diverse platforms and networks. Depending on your Tivoli Enterprise Portal configuration, you can collect and monitor data across platforms. The Monitoring Agent for IBM i gathers and filters status information at the managed resource rather than at the Hub, eliminating unnecessary data transmission and sending only data that is relevant to changes in status conditions. The Monitoring Agent for IBM i helps you monitor and gather consistent, accurate, and timely information that you need to effectively perform your job.

For detailed instructions about the installation, configuration, and usage of Tivoli Monitoring Agent for IBM i, refer to *IBM Tivoli Monitoring: i5/OS Agent User's Guide*, SC32-9448 found at:

http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/topic/com.ibm.itm.doc_6.2.fp1/main_os400.htm

6.8 Watch for Event function

The Watch for Event function enhances your ability to detect and react to problems. When specified messages, Licensed Internal Code log entries, or Product Activity Log entries occur, you are notified by calling a specified program to take the action you want.

The Watch for Event function is started by the Start Watch (STRWCH) command or the Start Watch (QSCSWCH) API to notify the user by calling an exit program when the specified event occurs. An event can be a message that is sent to a message queue, a job log, a Licensed Internal Code log entry, or a Product Activity Log (PAL) entry, which shows errors that occurred in disk and tape units, during communications, or on workstations.

Up to 10,000 watch sessions can be active at a time, but active watch session identifiers must be unique across the system.

The watch session continues until ended with the End Watch (ENDWCH) command or with the End Watch (QSCEWCH) API. A watch session can be ended from the same job or a different job.

6.8.1 Watch for Event categories

On IBM i V6.1, you can use the Watch for Event function to:

- Watch for messages
- Watch for Licensed Internal Code (LIC) log entries
- Watch for Product Activity Log (PAL) entries

> **Note:** You can specify up to five entries for each category when starting a Watch.

Watch for messages

When you watch for messages, specify the message queue or job log where you expect the message to be sent. You can narrow the search by specifying a text string to be compared against:

- Message data
- The From program of the watched-for message
- The To program of the watched-for message

Watch for LIC log entries

When you watch for Licensed Internal Code log entries, specify the Licensed Internal Code log major and minor codes. You can narrow the search by specifying a text string that is to be compared against:

- The number of the task dispatching element (TDE)
- The name of the task
- The type of server
- The name of the job
- The user name of the job
- The job number to further qualify the job name and user name of the job
- The thread identifier
- The exception identifier
- The LIC module name
- The LIC module replacement unit name
- The name of the entry point
- The byte offset into the LIC module text
- The time stamp of when the LIC module is compiled

Watch for PAL entries

When you watch for Product Activity Log entries, specify the particular system reference code (SRC) to be watched for. You can narrow the search by specifying a text string that is to be compared against:

- The name of the physical device that has the entry in the log
- The number or word that is used to identify a product
- The numbers or letters that are used to identify the feature level of a product with a given type

6.8.2 Watch for event commands and APIs

You can use either CL commands or APIs to work with the Watch for event function.

Start Watch (STRWCH)

The STRWCH command or QSCSWCH API starts a watch session and notifies you when a specified message, a Licensed Internal Code log entry, or a Product Activity Log entry occurs. When the watched-for message is added to the specified message queue or log, or when the watched-for log entry is added, the exit program specified in the *Watch program* (WCHPGM) parameter is called.

Work with Watches

The Work with Watches (WRKWCH) command can be used to show a window with the list of active watches on the system. An active watch can be a watch that was started by issuing Trace Internal (TRCINT), Start Trace(STRTRC), Trace Connection (TRCCNN), Trace TCP/IP Application (TRCTCPAPP), and Start Communications Trace (STRCMNTRC) commands specifying watches parameters, or a session that was started using the Start Watch (STRWCH) command, or using the Start Watch (QSCSWCH) API.

When you run this command, the Work with Watches window is shown. From this window, you can end an active watch or start a new one. Also the watches being ended are shown.

End Watch

The End Watch (ENDWCH) command ends a watch session that is started by the Start Watch (STRWCH) command or by the Start Watch (QSCSWCH) API. Watch sessions that are started by trace commands (such as STRTRC, TRCINT, TRCCNN, STRCMNTRC, TRCTCPAPP) are ended, but the associated trace remains active.

Note: To use the above commands, you must have service (*SERVICE) special authority, or be authorized with the Service watch function of IBM i through Application Administration support on either IBM Navigator for i or IBM Systems Director Navigator for i. The Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM_SERVICE_WATCH, can also be used to change the list of users that are allowed to start, end, or work with watch operations.

6.8.3 Additional information

For a sample scenario that uses the Watch for Event function with an exit program, refer to the IBM i V6.1 Information Center and select **Troubleshooting** → **Detecting problems** → **Watch for event function** → **Scenario: Using the watch for event function with an exit program**.

For more information about the Watch function, refer to the IBM i Information Center or Chapter 15, “Java Virtual Machine problem determination”, of *i5/OS Diagnostic Tools for System Administrators: An A to Z Reference for Problem Determination*, SG24-8253.

7

Trending and capacity planning tools

This chapter discusses the tools to use for trending and capacity planning purposes. the tools are Performance Management for Power Systems, IBM Systems Workload Estimator (WLE), and collecting, analyzing, and managing the data with self-developed tools.

© Copyright IBM Corp. 2009. All rights reserved.

7.1 The tools for trend analysis

The following sections provide you with a description of the available tools and show a short example of how those tools can be used.

7.1.1 Performance Management for Power Systems

IBM Performance Management (PM) for Power Systems is a helpful system management tool for supervising a system's performance. In this chapter, we explain how to get started with PM, discuss how to use it, and provide descriptions of the commonly used PM reports. In addition, we describe the interface to the WLE in 7.2.1, "IBM Systems Workload Estimator (WLE)" on page 187 and provide valuable information to help you manage and project your partition's growth.

Perform these steps:

1. Use the following URL to access the PM for Power Systems logon window that is shown in Figure 7-1:

   https://pmeserver.rochester.ibm.com/PMServerInfo/loginPage.jsp

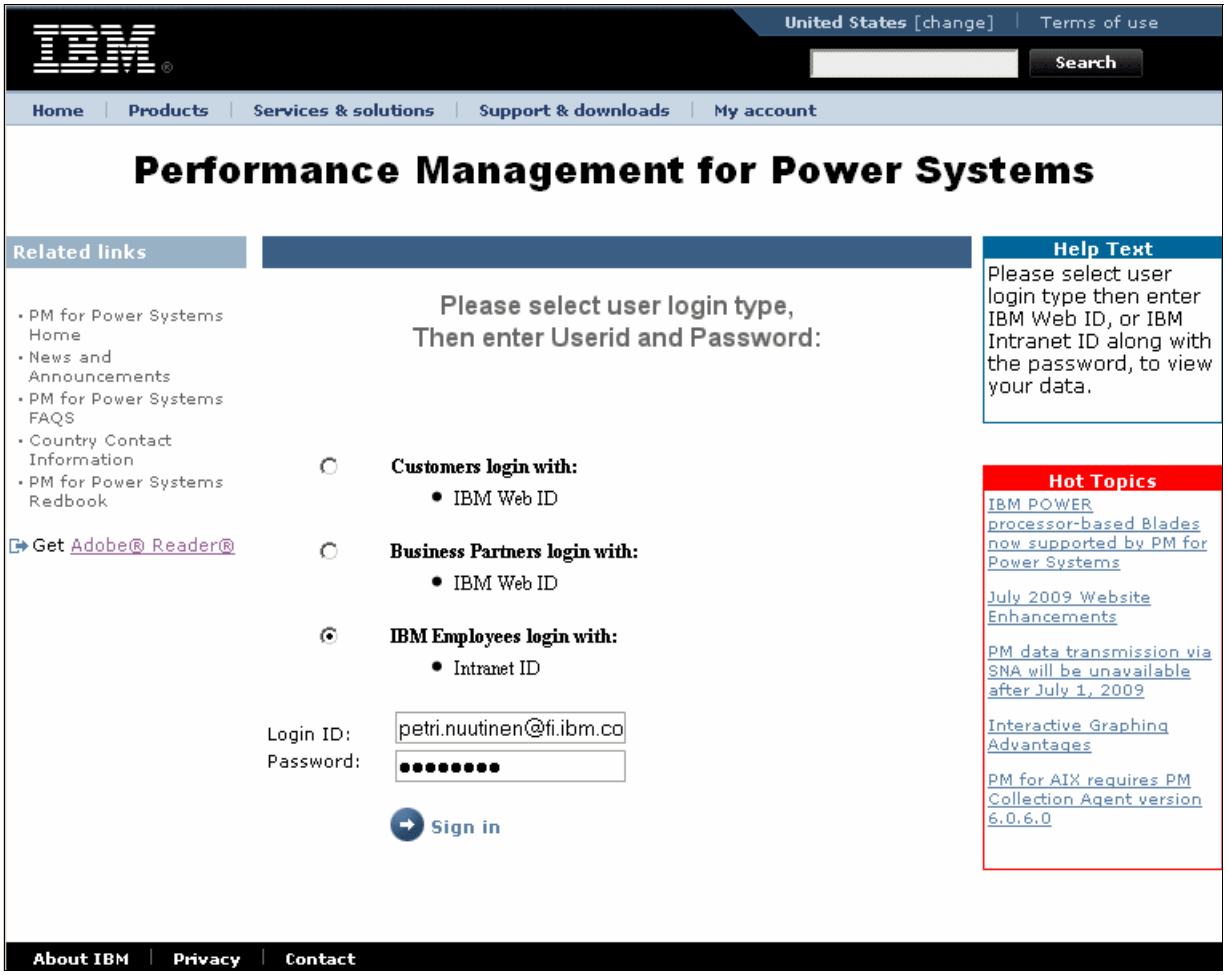


Figure 7-1 Sign on to Performance Management for Power Systems

2. Every time you start a session with PM for Power Systems, you need to accept the privacy acknowledgement shown in Figure 7-2.

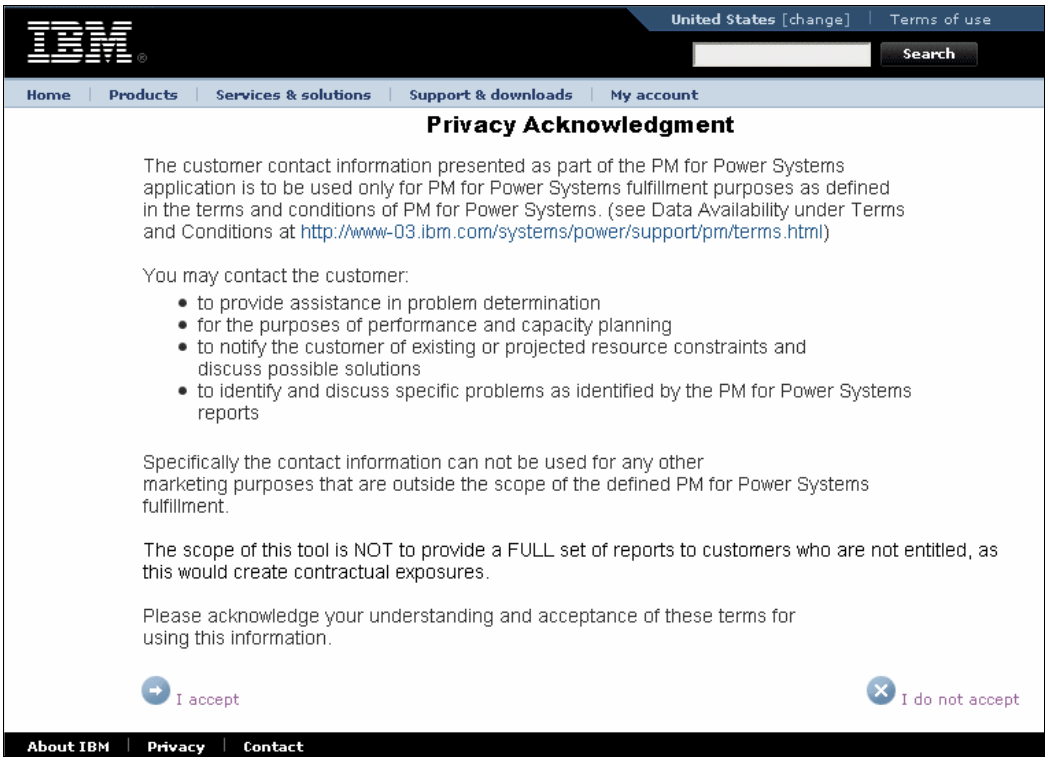


Figure 7-2 Privacy Acknowledgement for Performance Management for Power Systems

3. Select a partition or a group of partitions to work with, as shown on Figure 7-3. Scroll through the list of groups until the correct group is highlighted. Click the **Select Group** button.

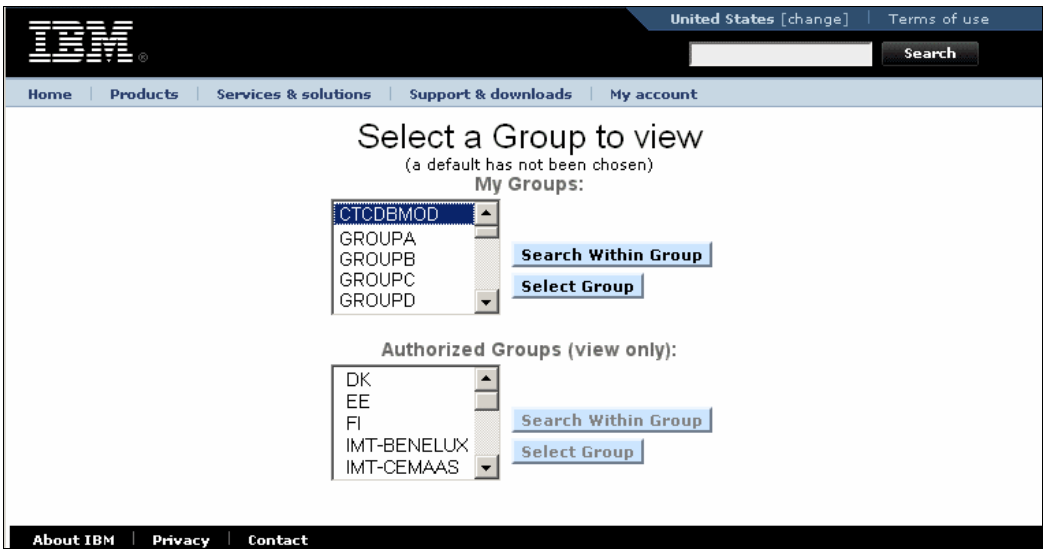


Figure 7-3 Select a Group to view data

The window shown in Figure 7-4 gives an example of showing the partitions within a server group.

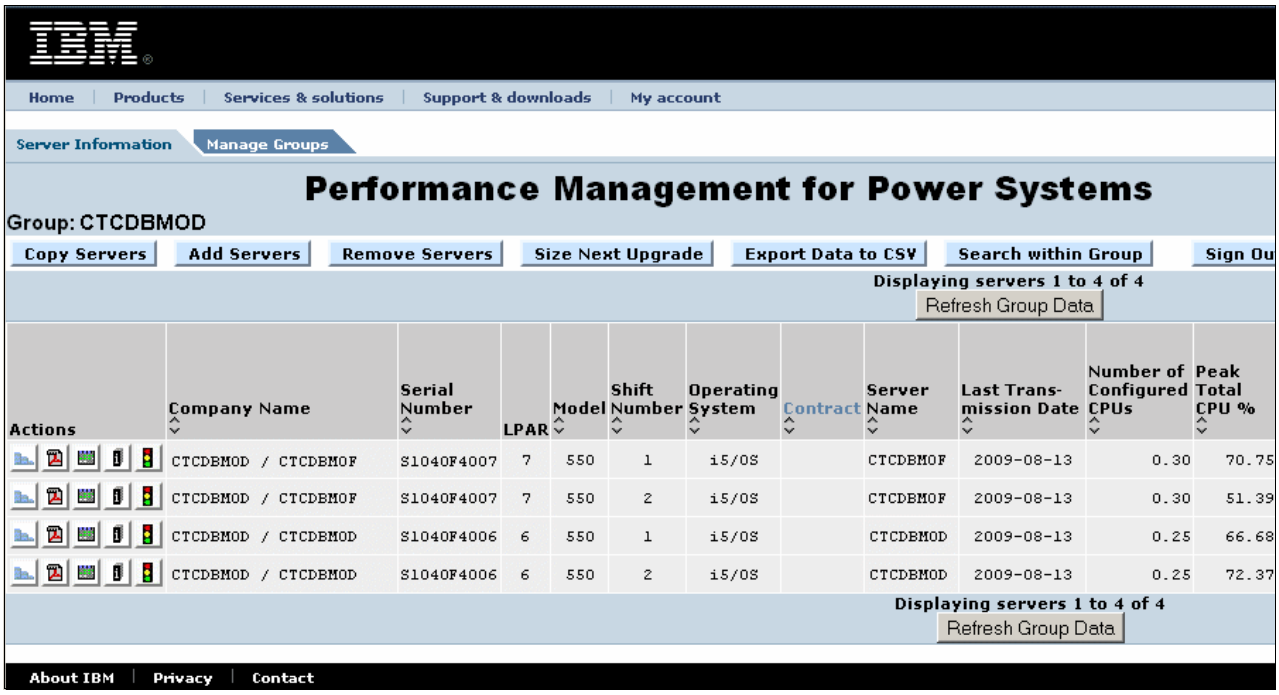


Figure 7-4 Showing the action buttons

On the left side of the window are the action buttons. Click the leftmost button on a row for detailed graphs. Select the appropriate tab to view the appropriate resource usage. Figure 7-5 provides an example of the report selection window.

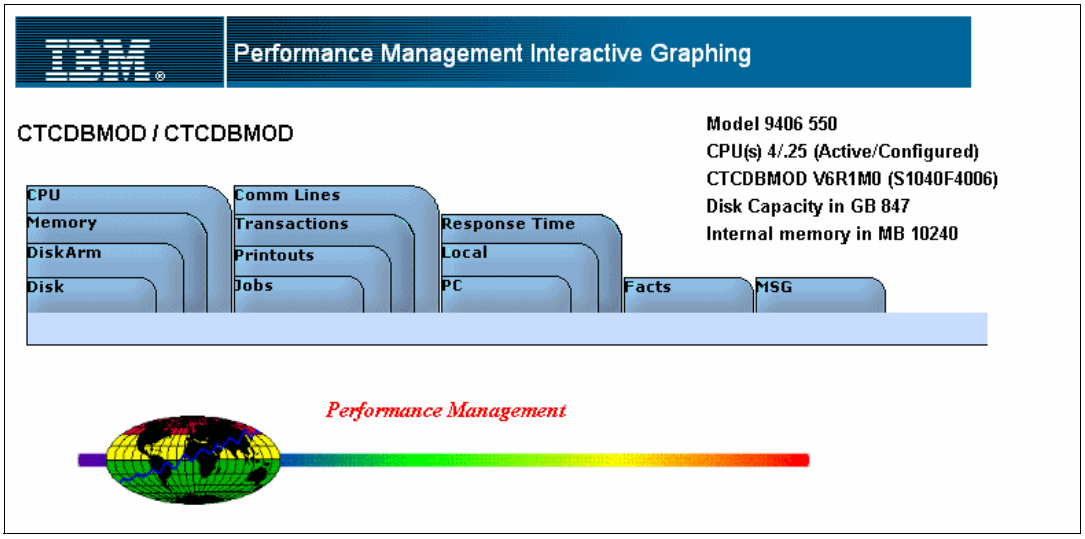


Figure 7-5 Selecting the report to work with

Selecting the **CPU** tab opens the window showing the CPU utilization during the last three months, as shown in Figure 7-6.

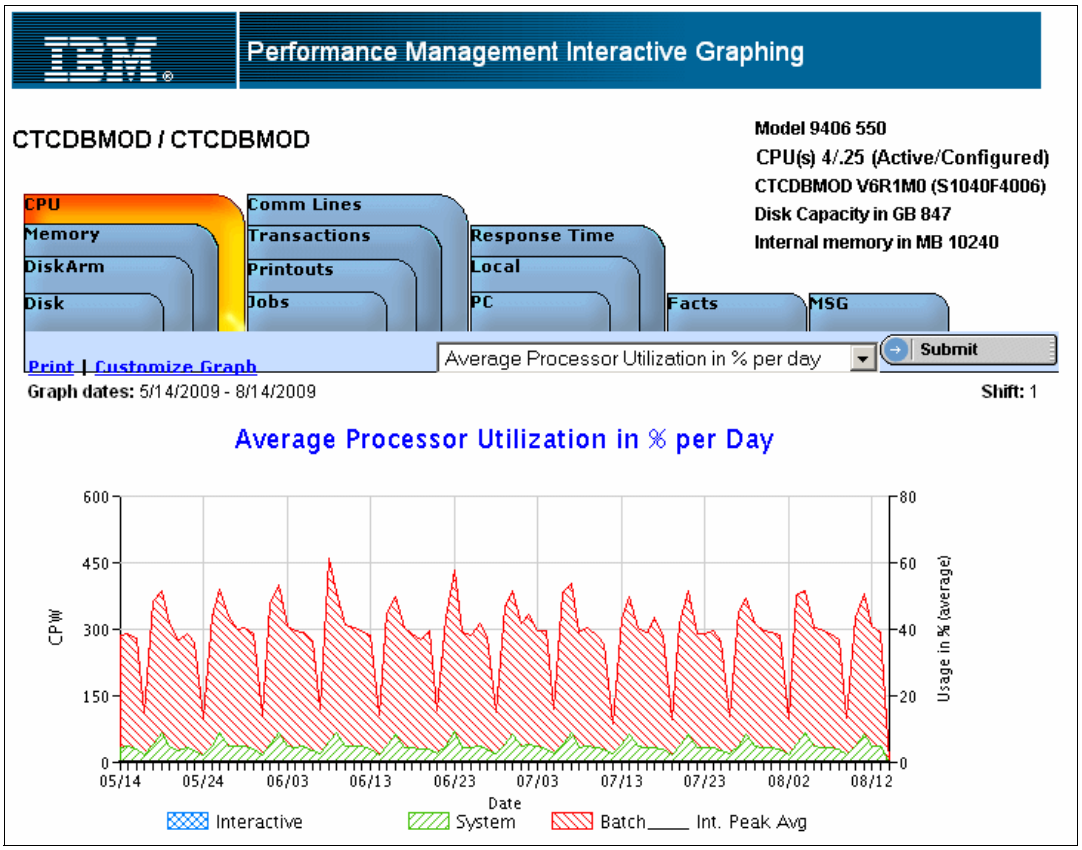


Figure 7-6 Average Processor Utilization during last three months

Select the drop-down menu and select the item you wish to inspect, as shown in Figure 7-7. Click the **Submit** button after selecting the option to retrieve the data from the server.

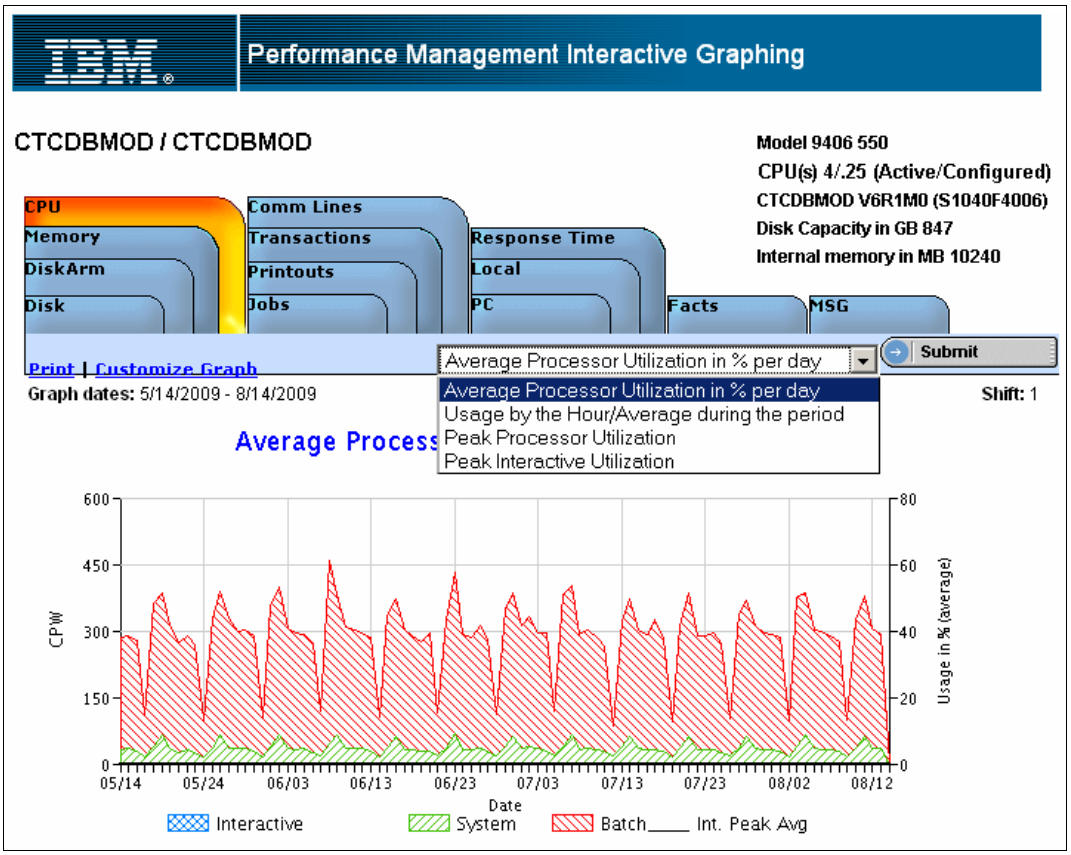


Figure 7-7 The graph options for the CPU reports

Select **Peak Processor Utilization** and click **Submit**. This provides up to a full year of measured data and three months of expected growth in the utilization graph. There are three versions of the growth data and they are calculated based on:

- The last three months of data
- The last six months of data
- The last twelve months of data

The graph that appears is quite large, so we show the upper part of the panel in Figure 7-8.

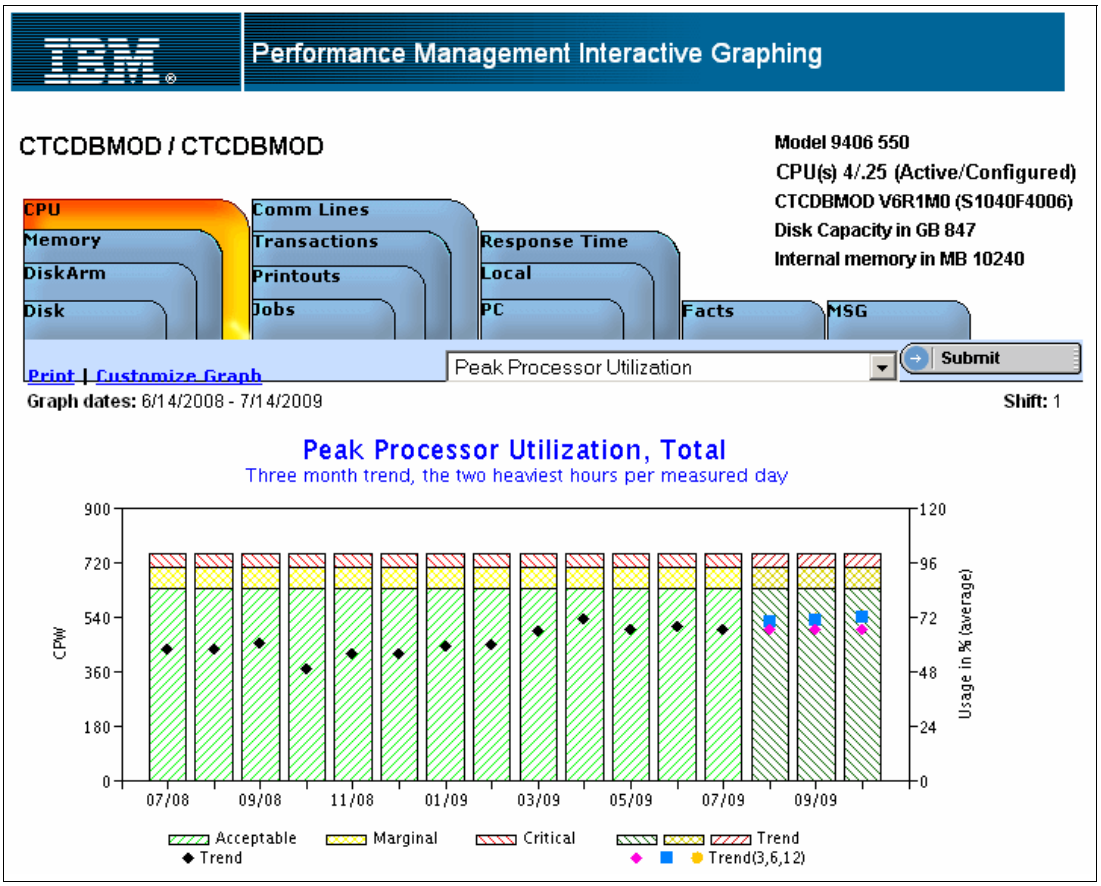


Figure 7-8 CPU utilization during last thirteen months

Provided that there is data collected for a full year, this graphics provide information about how the CPU utilization fluctuates within a period of one year. The black rectangle represents the CPU utilization during the two heaviest hours measured within any of the months displayed. The trend forecasts are shown on the three rightmost pillars.

The lower portion of the window shows both the guidelines and estimations, as shown in Figure 7-9.

Facts

	Peak	
	Marginal	Critical
Recommended guideline in %, interactive priority	48	62
Recommended guideline in %, system+int. priority	56	70
Recommended guideline in %, all priorities	85	94
Number of months until the guideline is reached, interactive priority	12	12
Number of months until the guideline is reached, system + int. priority	12	12
Number of months until the guideline is reached, all priorities	11	12

Figure 7-9 The fact sheet of the CPU utilization

Selecting the Disk Arm utilization history provides information about the disk arms. Figure 7-10 shows you an example of a sinking trend.

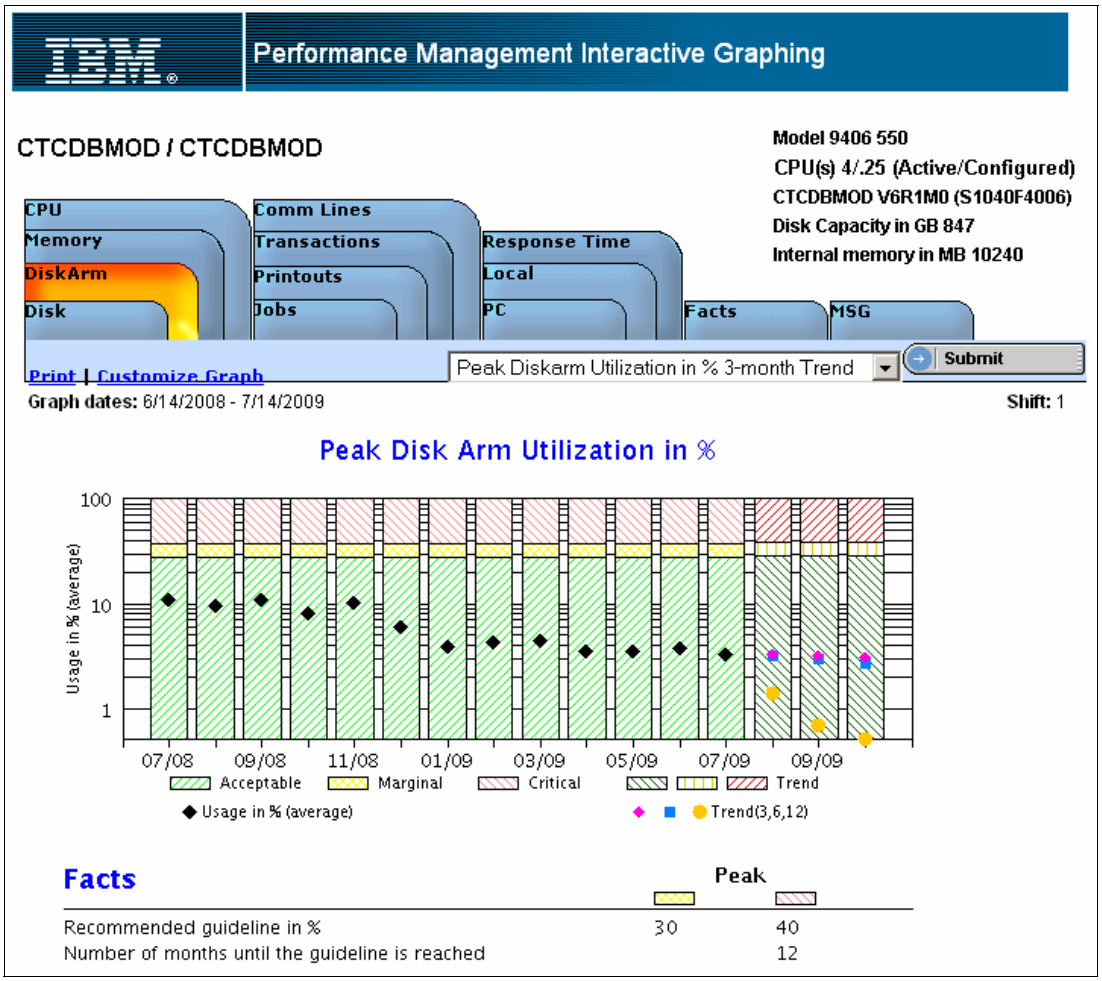


Figure 7-10 The Disk Arm Utilization history and trend

Besides Disk Arm Utilization, the Disk Space usage needs to be monitored, and an example of a graphic that shows the disk space usage during last thirteen months is shown in Figure 7-11.

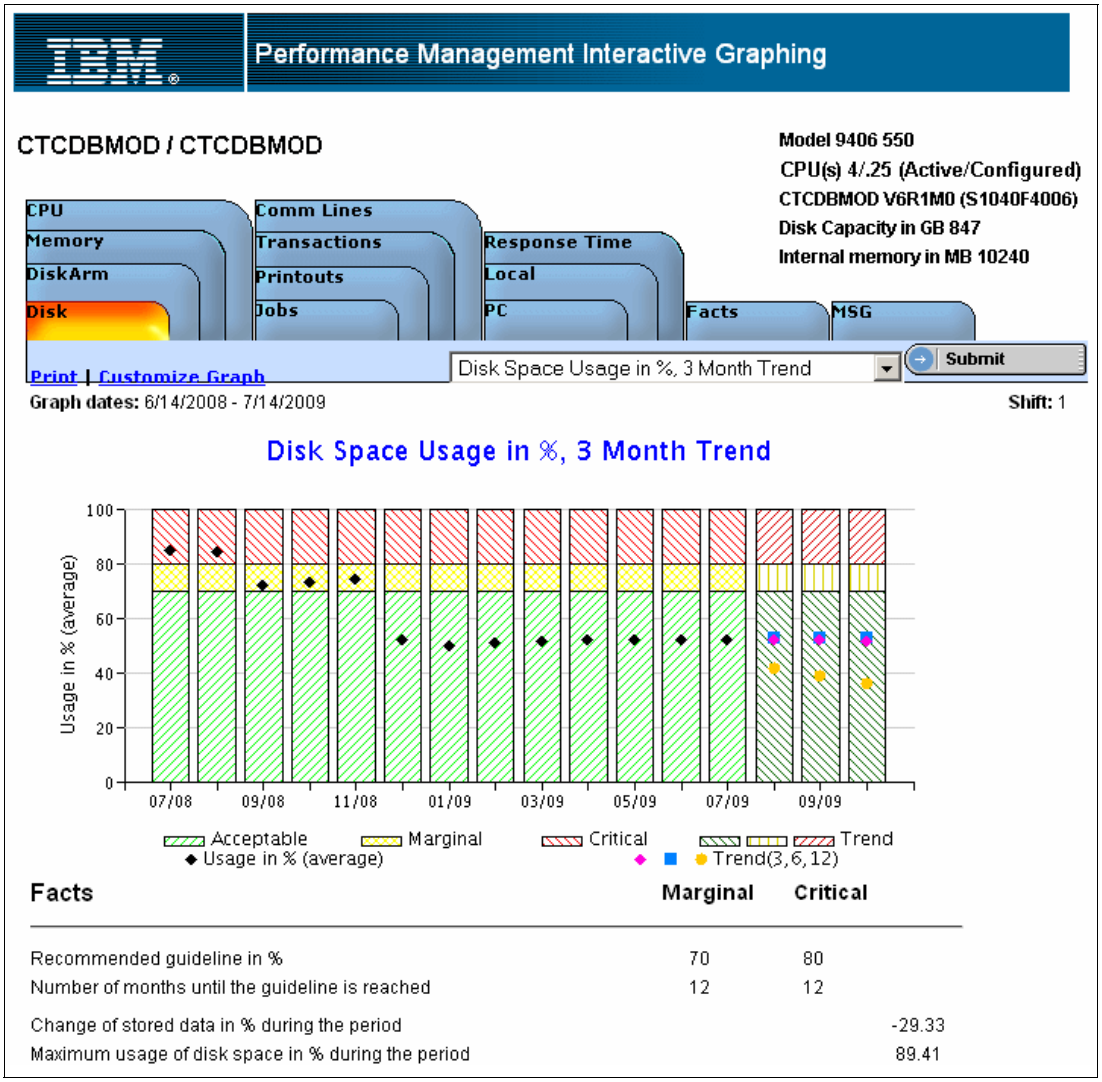


Figure 7-11 Disk Space Usage history and trend

These examples give you a good idea about how useful PM for Power Systems is. For more information about how to use this tool, refer to *A Systems Management Guide to Performance Management for System i and System p servers*, SG24-7122. To view the book online, go to the following address:

http://www.redbooks.ibm.com/abstracts/sg247122.html?Open

7.1.2 Using Navigator for i for trending

A graph history provides a graphical view of performance data collected over days, weeks, months, or even years with Collection Services. You do not need to have a system monitor running to view performance data. As long as you use Collection Services to collect data, you can view the Graph History window.

The amount of data that is available to be graphed is determined by the settings that you selected from the Collection Services properties (specifically the collection retention period).

> **Tip:** You can use the Navigator for i to activate the Performance Management Agent (PM Agent) over multiple systems.

When you activate the PM Agent, you can use the graph history function to see data that was collected days, weeks, or months ago. You can go beyond the real-time monitor capabilities, and have access to a summary or detailed data. Without PM Agent enabled, the graph data field supports 1 to 7 days. With PM Agent enabled, you define how long your management collection objects remain on the system. For example:

- Detailed data (attribute type *PFR in QMPGDATA.LIB or QPFRDATA.LIB)

  The length of time that management collection objects remain in the file system before they are deleted. You can select a specific time period in hours or days, or you can select Permanent. If you select Permanent, the management collection objects will not be automatically deleted.

- Graph data (attribute type *PFRDTL in QMGTC2.LIB)

  The length of time that the details and properties data that is shown in the Graph History window remains in the system before it is deleted. If you do not start PM Agent, you can specify 1 to 7 days. If you start PM Agent, you can specify 1 to 30 days. The default is 1 hour.

- Graph history data (attribute type *PFRHST in QMGTC2.LIB)

  The length of time that the data collection points of a graph can be displayed in the Graph History window or remain in the system before they are deleted. No details or properties data is available. You must start PM Agent to enable the summary data fields. The default is one month. The summary data is summarized in 1 hour intervals and does not support second- and third-level details.

- Graph history status

  The Graph History window now displays the graph history status. You also can recreate the graph history data if it is missing.

How to use the Navigator for i historical graphs

Perform these steps:

1. Start Navigator for i and sign on to the System i partition that you want to inspect. Open the **Monitors** tab, right-click **System**, and select the **Graph History**, as shown in Figure 7-12.

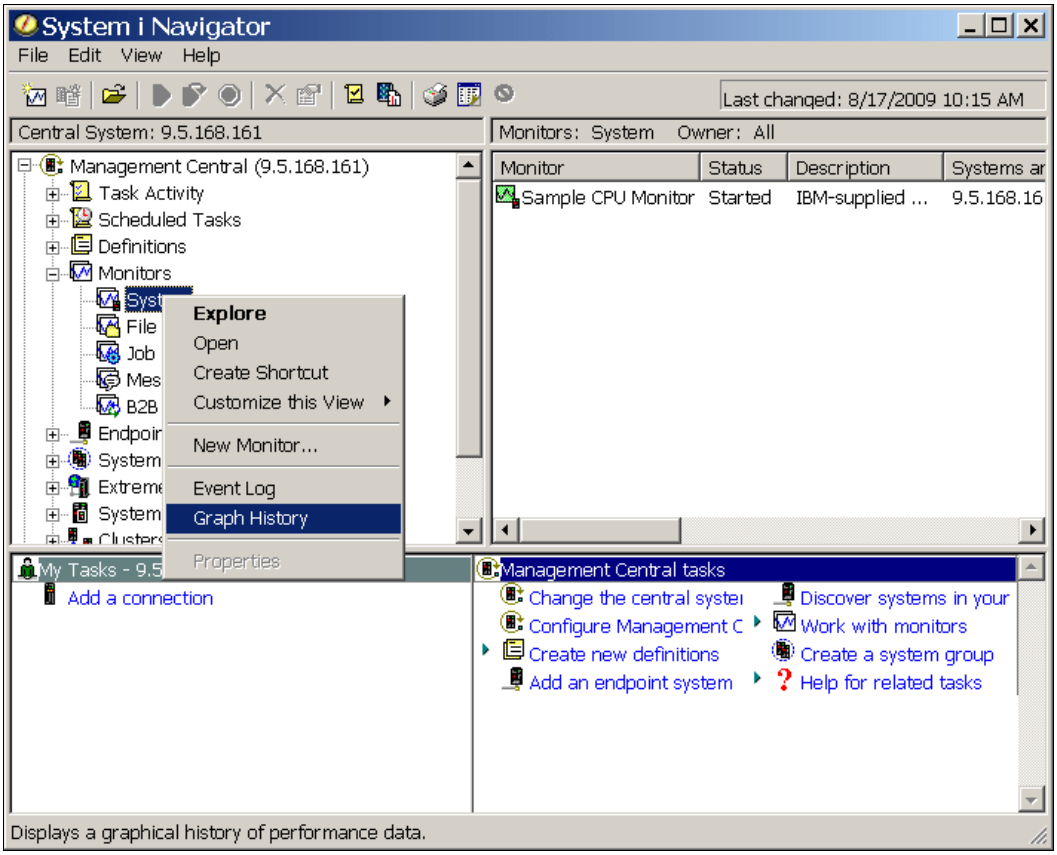


Figure 7-12 Starting the Graph History function

2. Specify the dates and the amount of granularity you want to investigate. In Figure 7-13, we specify to investigate a CPU utilization history of one week by making selections in the drop-down menus.

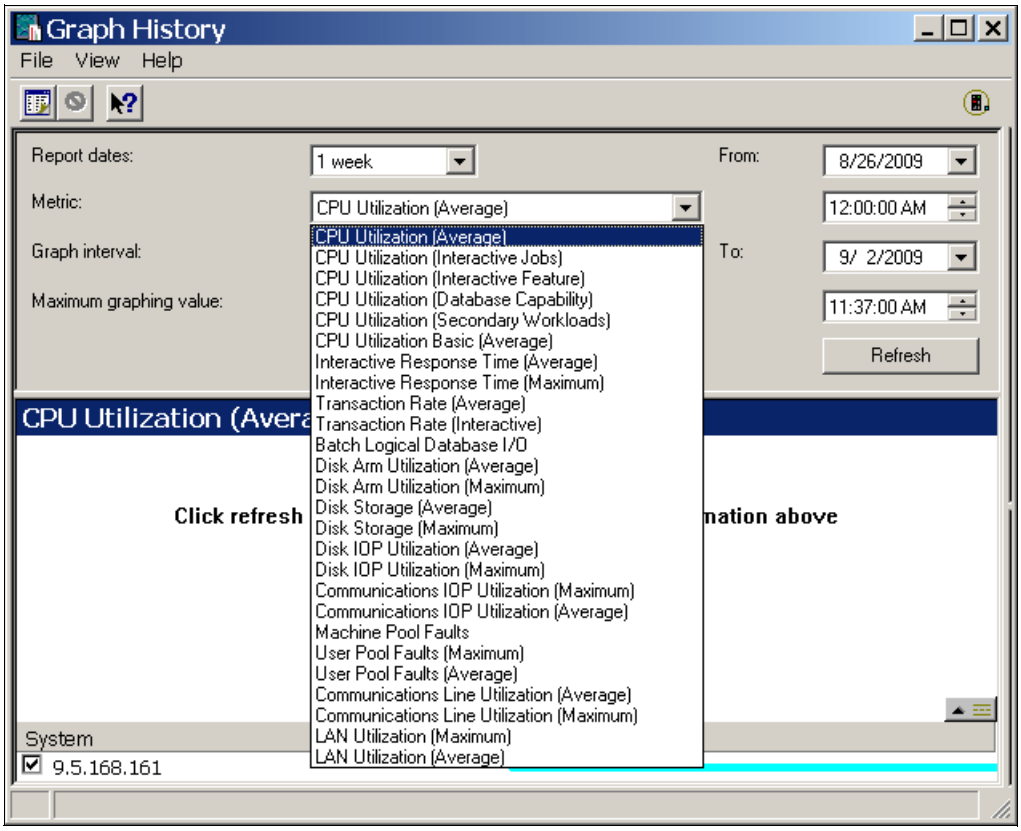


Figure 7-13 Specifying the metrics for the History Graph

3. Click the **Refresh** button. If the history to investigate exceeds seven days and the PM Agent was not activated, you must start it, as shown in Figure 7-14.

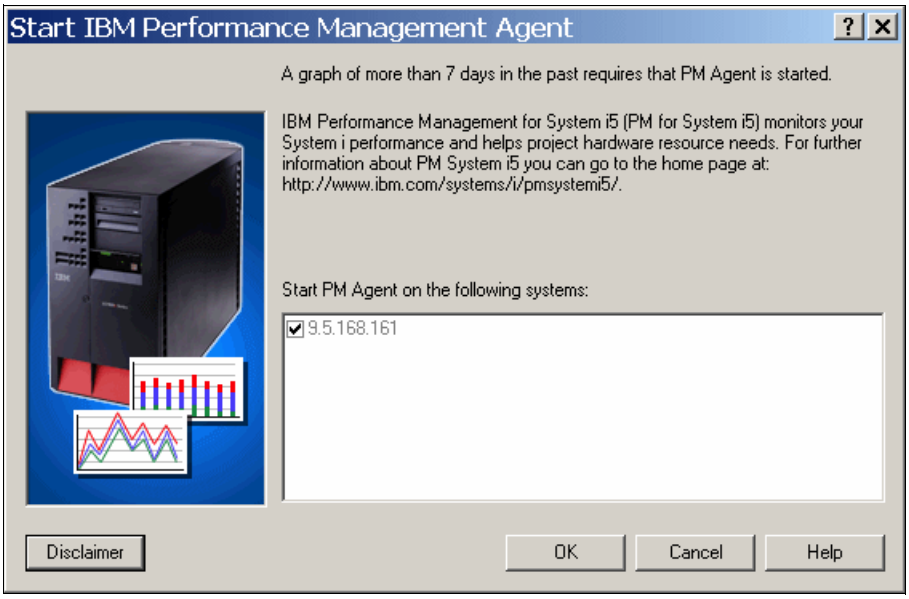


Figure 7-14 Start the Performance Management Agent

4. A disclaimer appears, as shown in Figure 7-15.

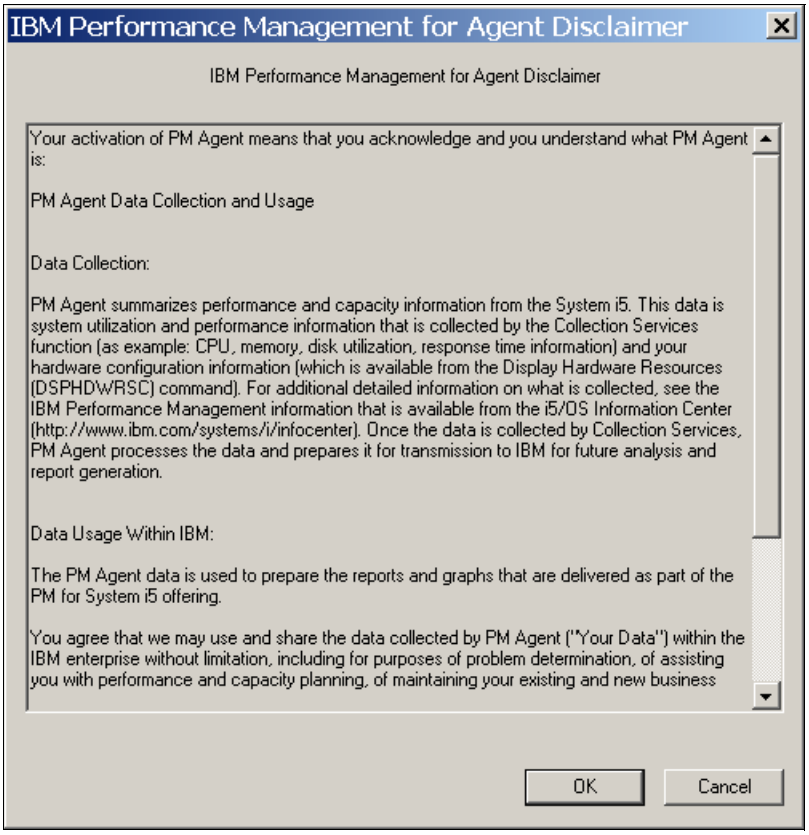


Figure 7-15 The Agent Disclaimer

5. Click **OK**. A window opens, where you can select the metrics to display, as shown in Figure 7-16.

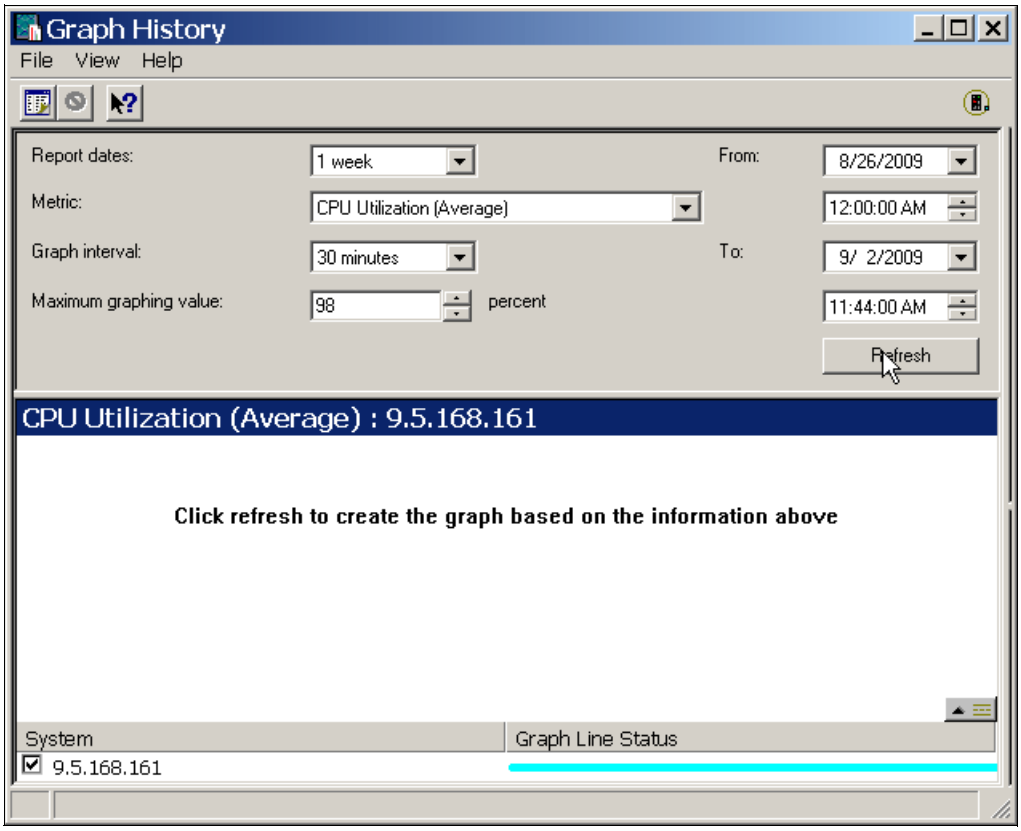


Figure 7-16 Accepting the metrics for Graph History

6. Verify that the parameters are correct and click **Refresh**. The window shown in Figure 7-17 opens.

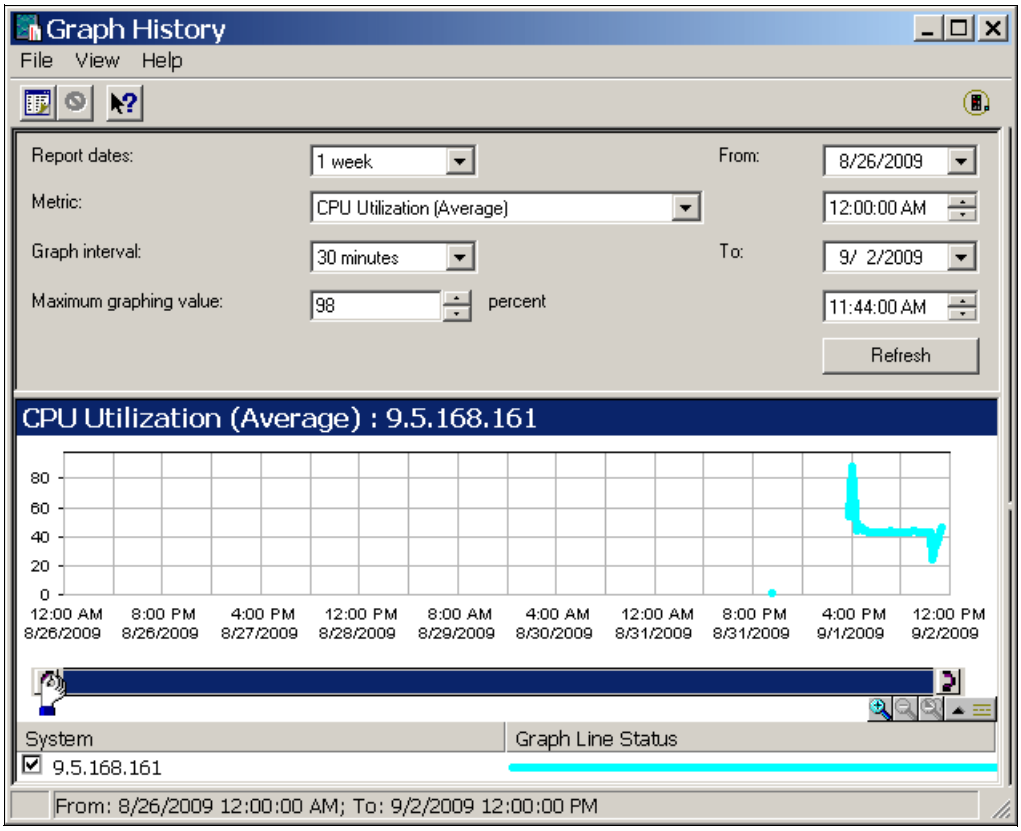


Figure 7-17 One week of CPU Utilization history graph

In this example, the graph starts from the end of the timeline, as there is no workload on the partition from where this performance data is collected.

7. Grab the left hand side of handle at the bottom of the window and pull it to the right to expand the timeline closer to the current time. An example of the CPU utilization is shown in Figure 7-18.

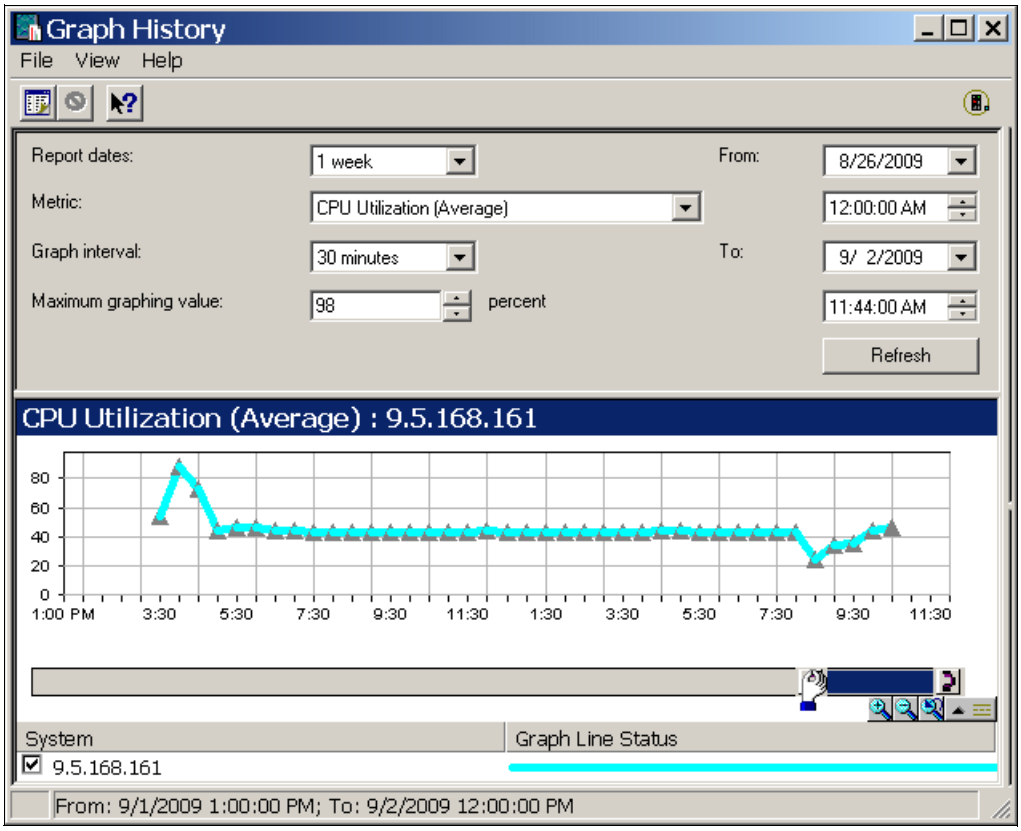


Figure 7-18 The CPU history with a shorter timeline

8. As an example, we show how the memory has been used during the same time frame. Select **User Pool Faults (Maximum)** in the drop-down menu and click **Refresh**. The resulting graph is shown in Figure 7-19.

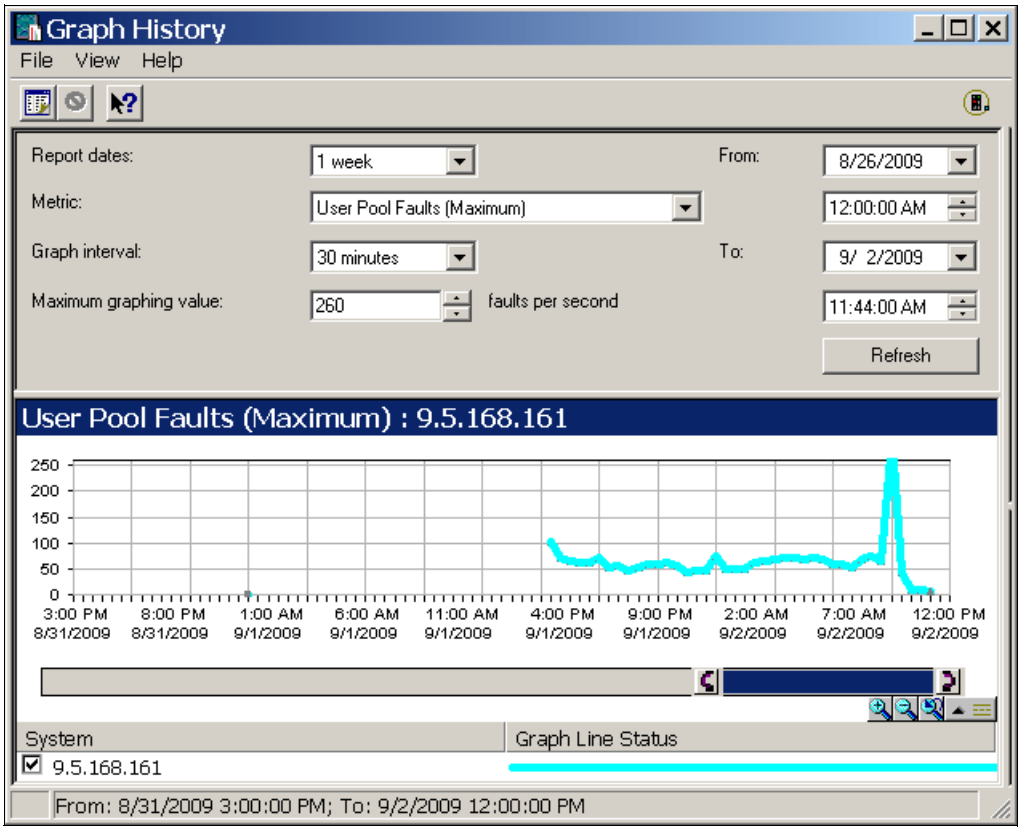


Figure 7-19 The User Pool Faults history graph

Please note that you are responsible for finding the trends while using these graphs; there is no trending calculation function on these graphs, as opposed to the graphs produced by the Performance Management for Power Systems service offering.

7.1.3 Using CL commands for trend analysis

This method take more time and effort, as you are responsible for creating all the data for trend analysis purposes. The steps you need to perform are:

1. Collect data regularly and create performance data by running the CRTPFRDTA command.
2. Copy the data to the historical data library by running the CPYPFRDTA or CPYPFRCOL command.
3. Produce the graphs by running the DSPPFRGPH command.
4. Analyze the graphs.

CL commands for collecting the data for trending

There are some command language commands that you may use for both analyzing the partition's performance in real time and directing the output to a spool file for a later analysis.

WRKACTJOB

The Work with Active Jobs (WRKACTJOB) command allows you to work with performance and status information for the active jobs in the system. The sequence of jobs can be changed with the Sequence (SEQ) parameter or through operations on the display. Other parameters allow the selection of jobs to be shown on the display. The selection parameters can also be changed by operations on the display.

WRKDSKSTS

The Work with Disk Status (WRKDSKSTS) command allows you to display and work with performance and status information for the disk units on the system.

WRKSYSACT

The Work with System Activity (WRKSYSACT) command allows you to interactively work with the jobs and tasks currently running in the system. Besides having the capacity to view this data on the display station, you may also direct the data to be stored in a database file for future use.

WRKSYSSTS

The Work with System Status (WRKSYSSTS) command allows you to work with information about the current status of the system. It displays the number of jobs currently in the system, the total capacity of the system auxiliary storage pool (ASP), the percentage of the system ASP currently in use, the amount of temporary storage currently in use, the percentage of machine addresses used, and statistical information related to each storage pool that currently has main storage allocated to it. The statistical information is gathered during an identified time interval (shown as the elapsed time); the data can either be updated by extending the measurement time interval, or it can be restarted to show statistics for the interval starting with the previous display.

Performing the trend analysis with CL commands

IBM provides commands for collecting performance data, managing performance data, and displaying performance graphics.

Create Performance Data command

The Create Performance Data (CRTPFRDTA) command creates a set of performance database files from performance information stored in a management collection (*MGTCOL) object. For more information about the database files, see the Performance topic in the Information Center.

Performance database files and file members will be created as needed based on the data contained in the management collection object and the information requested on this command. If database files already exist and the requested member exists in any of them, the member will be cleared before the collection is generated.

You may create the performance data files before you can copy them to the library where you keep the data for trending purposes. You may also copy the management collection objects to another library. See 4.2.4, “Managing Collection Services data” on page 58 for more information on this functionality.

Copy Performance Collection command

The Copy Performance Collection (CPYPFRCOL) command creates a copy of a performance collection object.

Display Performance Graph command

The Display Performance Graph (DSPPFRGPH) command produces a graph based on the performance data.

The graph format must have been defined by the Create Graph Format (CRTGPHFMT) command. The graph can be sent as output to a graphical terminal, non-graphical terminal, printer, plotter, and a graphics data format (GDF) file that can be used by other systems. Jobs can be selectively included or omitted from the graph.

Analyze Performance Data command

The Analyze Performance Data (ANZPFRDTA) command produces recommendations to improve the performance of the partition from which the data was collected. In interactive mode, you can request that the application make the recommended changes. In batch mode, the recommended changes are output, and you must then enter the individual commands to make the recommended changes.

Note: The ANZPFRDTA command analyzes one data set at a time. It is not a trending tool.

Compare the data with baseline

As trend analysis is part of the performance management life cycle, you need to compare the most recent performance data with the earlier created baseline. Unfortunately, there are no CL commands for that task.

Some trends you need to consider:

- Is the CPU utilization the same as or lower than the baseline?
- Are the number of transactions above or below the baseline?
- Is the amount of disk space used the same or is it growing?

We provide some utilization guidelines that should help you to decide when the resource usage is getting close to the point where you should start performing capacity planning functions. These guidelines are described in 8.2, “Resource utilization guidelines” on page 221.

7.2 Tools for capacity planning

The IBM tool for capacity planning is the Workload Estimator (WLE). This section will show you how to enter a workload into the WLE tool and build a hardware configuration that is able to support the workload.

7.2.1 IBM Systems Workload Estimator (WLE)

There are three methods of using the WLE for sizing:

- Input the existing data from the PM for Power Systems.
- Input data from Performance Data Investigator.
- Collect a new set of sizing data.

Using data from PM for Power Systems

Follow this procedure when using PM for Power Systems data as input:

1. Sign on to the PM, as shown in 7.1.1, “Performance Management for Power Systems” on page 170.

2. Select the group to work with and select the server to start sizing a new system by clicking the **Size Next Upgrade** button, as shown in Figure 7-20.

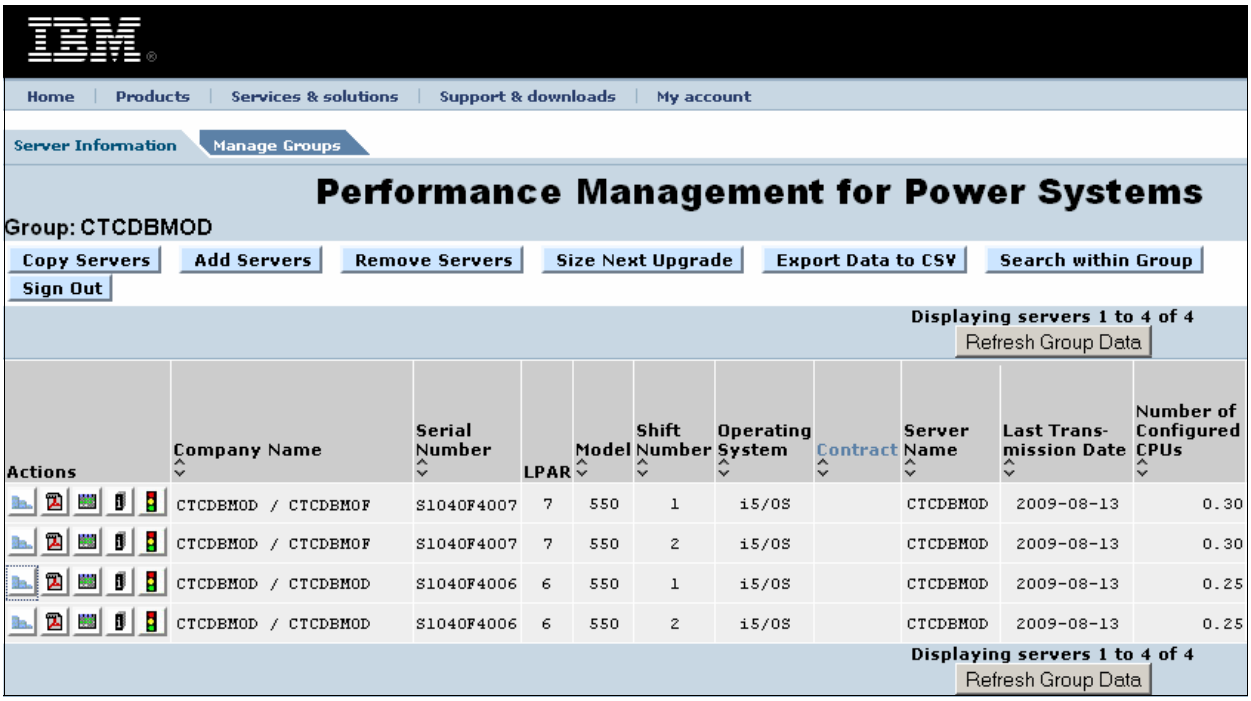


Figure 7-20 Size Next Upgrade button

3. Clicking **Size Next Upgrade** produces a new column of check boxes, as shown in Figure 7-21. As an example, we select one partition only, but you can use this same procedure to consolidate multiple partitions.

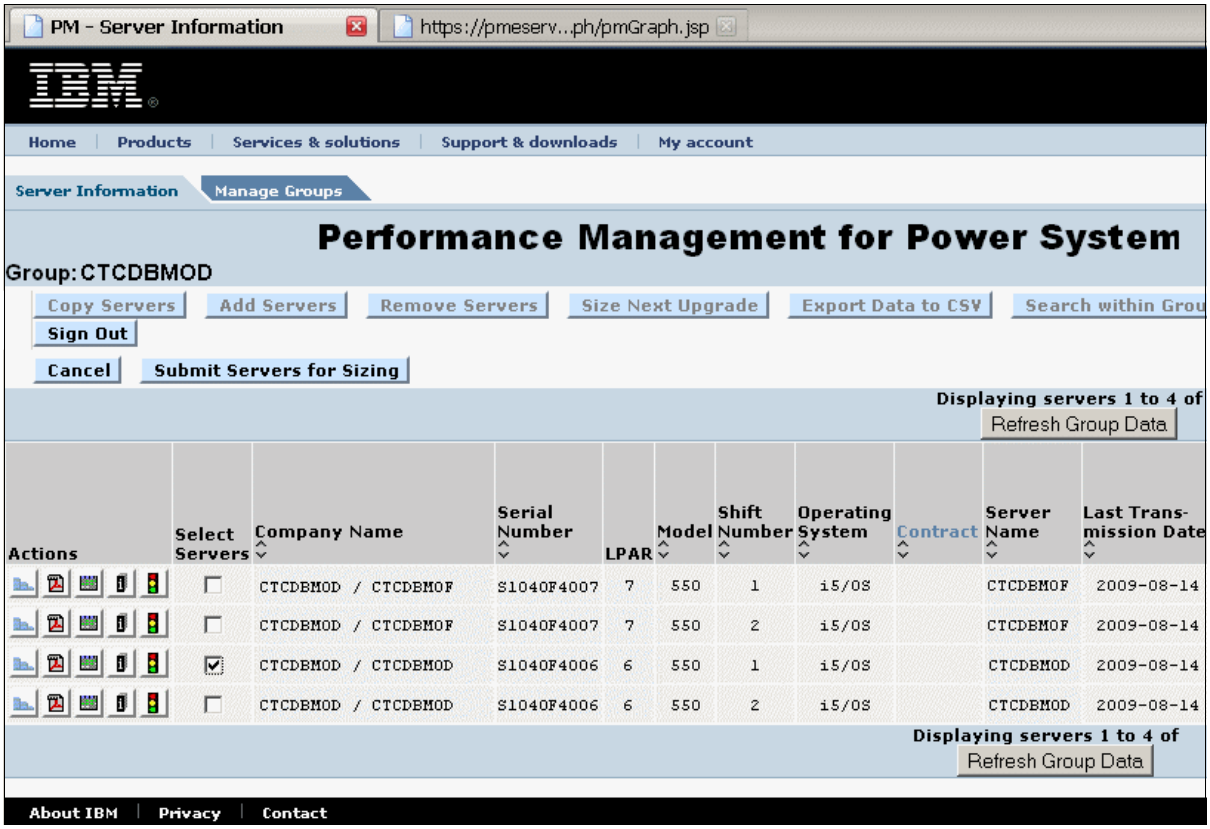


Figure 7-21 Selecting a server for sizing

4. Click the **Submit Servers for Sizing** button and a confirmation window opens, as shown in Figure 7-22.

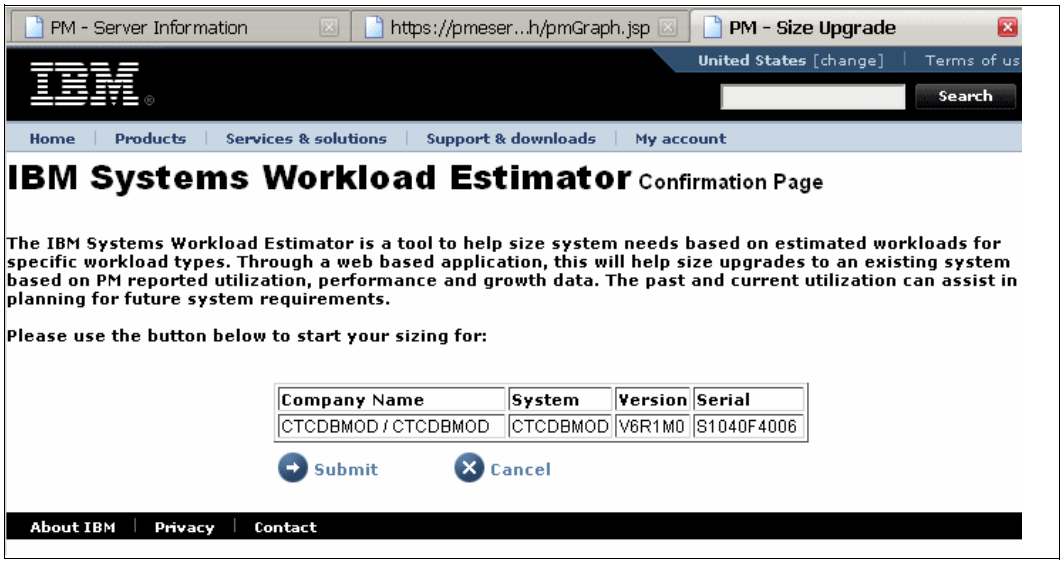


Figure 7-22 Confirming the selected partition

5. Click the **Submit** button and the workload definition window opens, as shown in Figure 7-23. The window is actually a combination of windows; the upper portion provides information about the available CPU, memory, and disk resources. It also shows the operating system level on which this server is running. The bottom portion of the window shows the growth information based on the data that is transmitted to the server.

Related Links
- IBM Systems Energy Estimator for POWER6 models
- Power Load Calculator for POWER5 models
- IBM System x and BladeCenter Power Configurator

Workload selection | Workload definition | Selected system | Help/Tutorials

→ 10-40F4006

→ User options
→ Reset this workload
→ Refresh this workload
Save this workload
→ Edit workload name
→ Modify intervals

Partition: CTCDBMOD, IBM i - 6.1, LPAR Shared Processor Capped

1. System Information:
60 weeks of data available.
Most recent entry:2009-08-02

Company Name:	CTCDBMOD / CTCDBMOD
Serial:	10-40F4006
Model:	550-8958/7463
Feature:	7463
System CPU:	66.8 % of 825 CPW
5250 OLTP CPU:	0.0 % of 750 CPW
Number of Cores:	0.25
Operating System:	IBM i - 5.4
Memory:	10,240 MB installed

Disk Group Info:

Group	Busy (%)	Units	Used (GB)	Total (GB)
ASP1	9	24	446	847

2. Will this workload be changed to use WebFacing or HATS Support?
- WebFacing
- HATS
- No

3. Growth Information:

Months to Grow:	12		Memory Growth Matches:
Interactive CPU:	0.068 CPW/Month		
NonInteractive CPU:	7.06 CPW/Month		
Total System CPU:	7.129	/Month	
Memory:	88.482 MB/Month		(Grow Independently)

Disk Group Growth Info:

Group Name	Storage Used(GB)	Inferred Disk Arm I/O's Consumed	Read Ops	Read IOSize (bytes)	Write Ops	Write IOSize (bytes)
ASP1	-1.17	N/A	-24.13	490.32	-3.29	27,251.65
ASP-001	0.00	0.00	N/A	N/A	N/A	N/A

4. Select a Virtual IO Server to provide Virtual Ethernet support: None
5. Select a Virtual IO Server to provide Virtual SCSI support: None

Back | Advanced Growth Options | Detailed Data | Continue

Figure 7-23 The Workload definition window

6. Clicking the **Advanced Growth Options** button allows you to specify different growth factors and to select only the desired periods for growth analysis. An example of this function is shown in Figure 7-24.

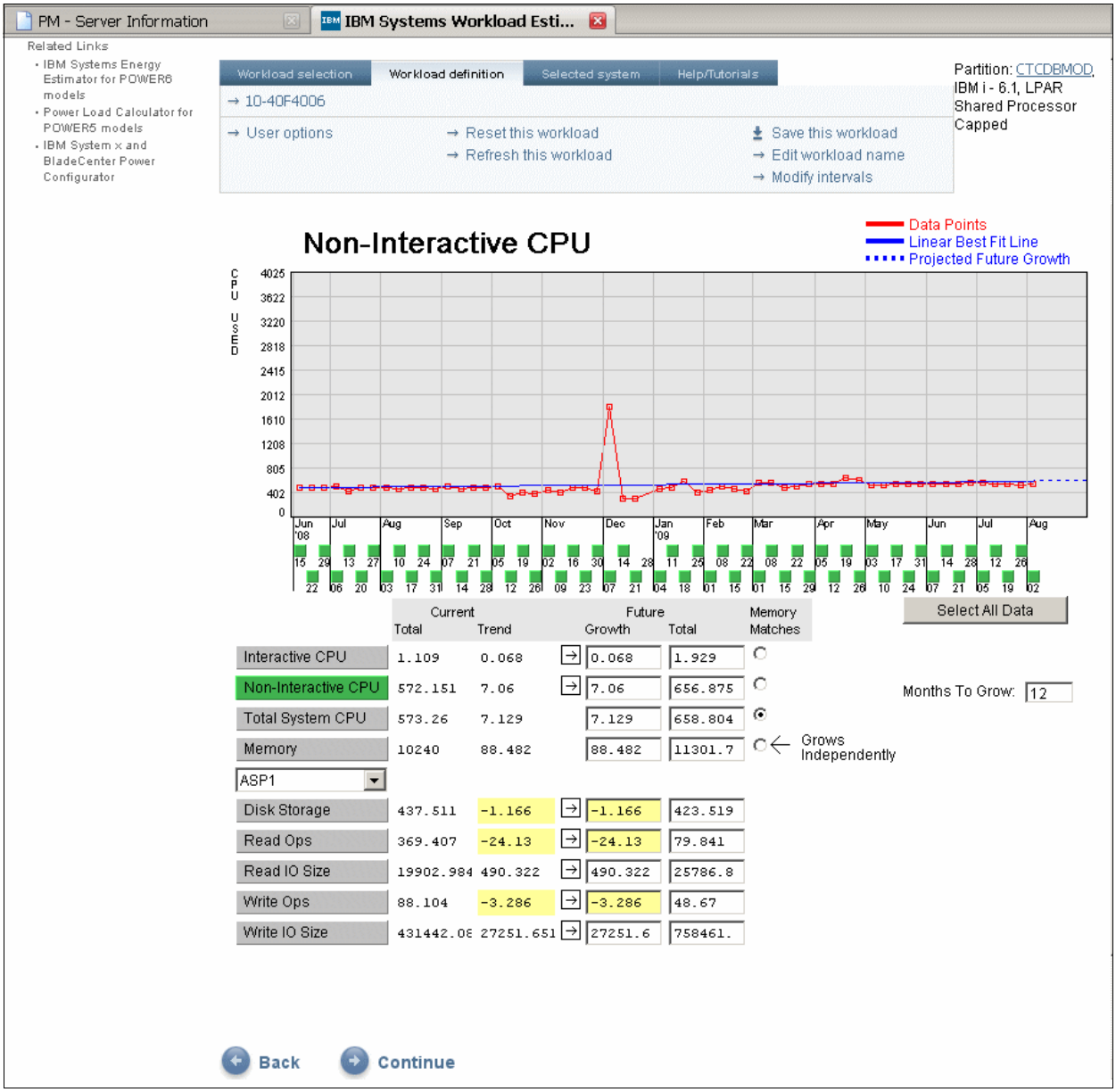


Figure 7-24 Advanced Growth Options window

7. As we do not want advanced functions in this example, we click the **Back** button to return to the Workload Definition window. Click the **Continue** button. The Selected System window opens, as shown in Figure 7-25, where we choose the base system for upgrading.

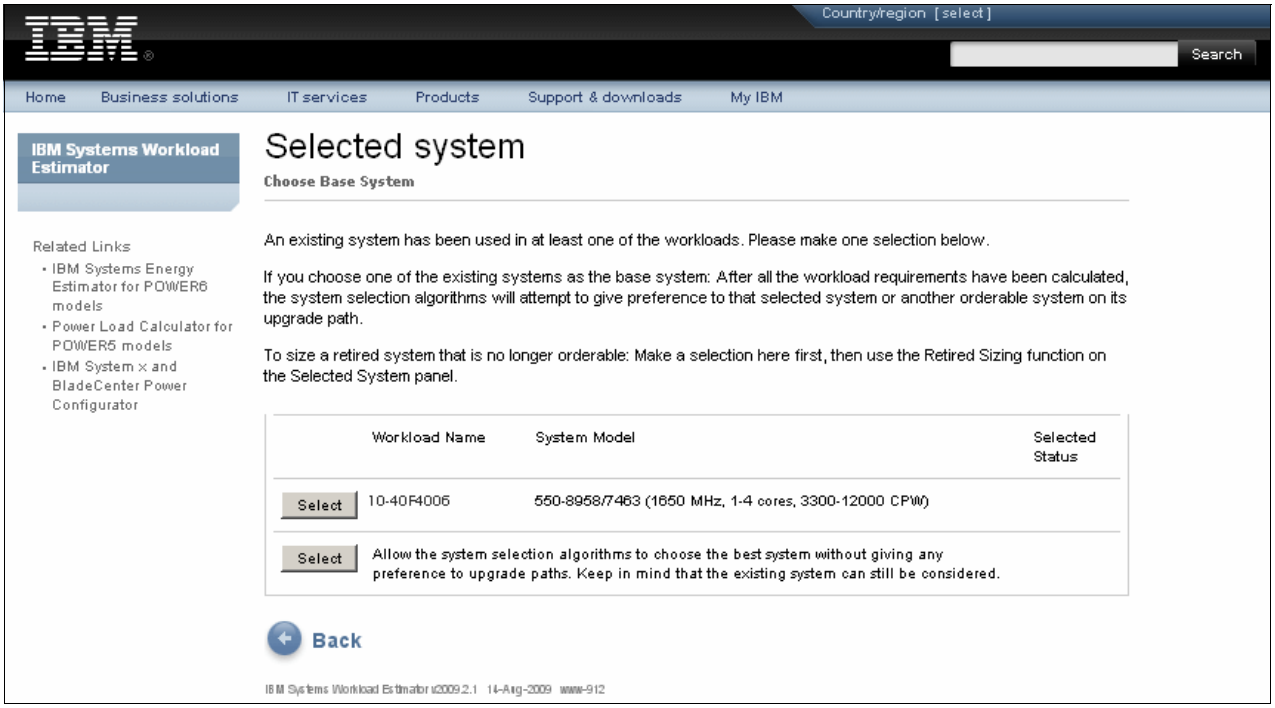


Figure 7-25 Choosing the base system

8. Selecting one of the two options opens the Selected System window that shows the recommended hardware for both the Immediate and Growth solutions. An example of this is shown in Figure 7-26.

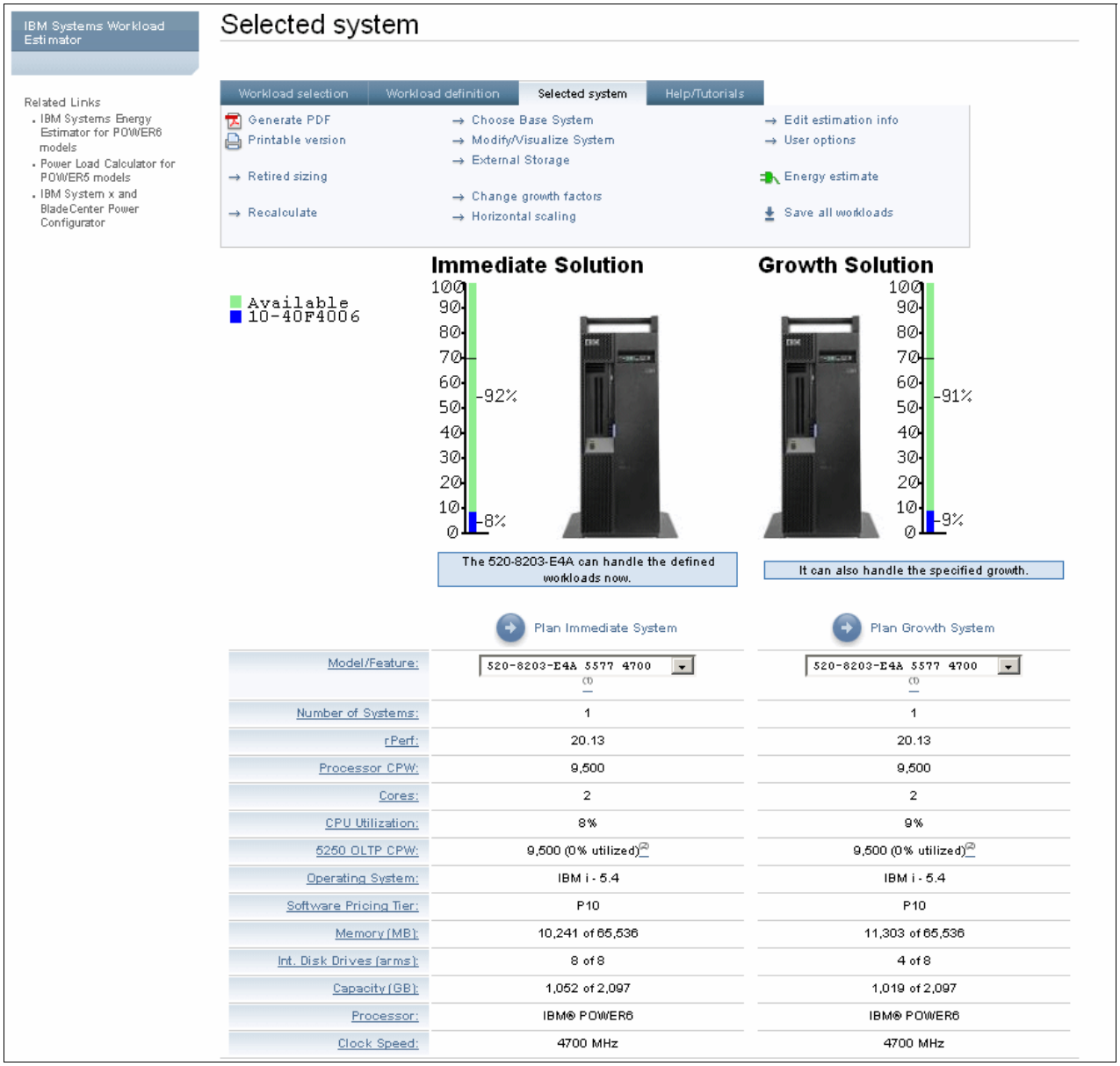


Figure 7-26 The Selected Systems window

Starting the Workload Estimator from Performance Data Investigator

The second way of importing an existing workload to the WLE is to import the data from the Performance Data Investigator (PDI). The following steps show you how to do this task. Please note that the Disk Health is not the only perspective that can be used to perform the Size Next Upgrade action on the PDI. We select this perspective for illustrative purposes only; you may use any of the perspectives to enter the data from PDI.

> **Note:** Size Next Upgrade is only available from a perspective that uses Collection Services data. The other perspectives with Job Watcher/Disk Watcher/PEX do not allow this option.

Perform these steps:

1. Open IBM Systems Director Navigator for i, as discussed in “Accessing the Investigate Data window” on page 82. Navigate to the Performance tab and select the **Performance** menu, as shown in Figure 7-27.

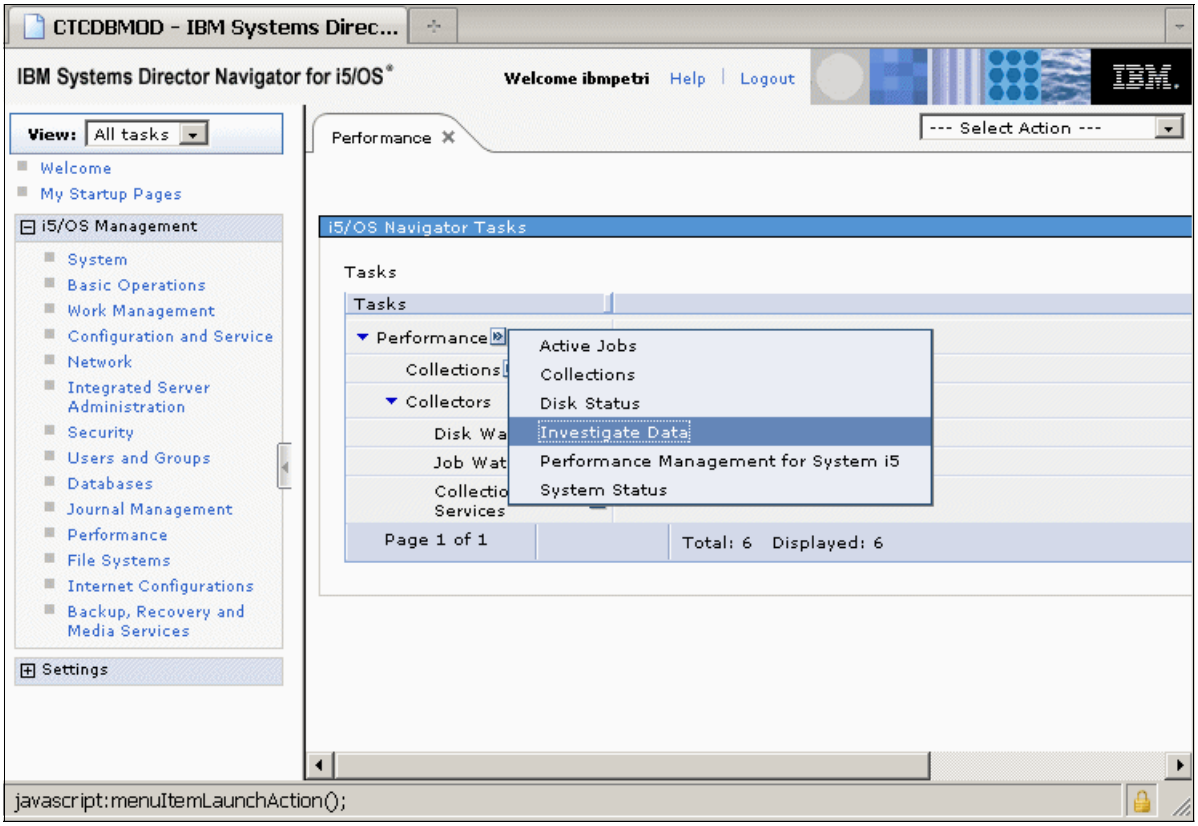


Figure 7-27 Starting the Performance Data Investigator

2. Select the **Investigate Data** task, which opens the perspectives shown in Figure 7-28.

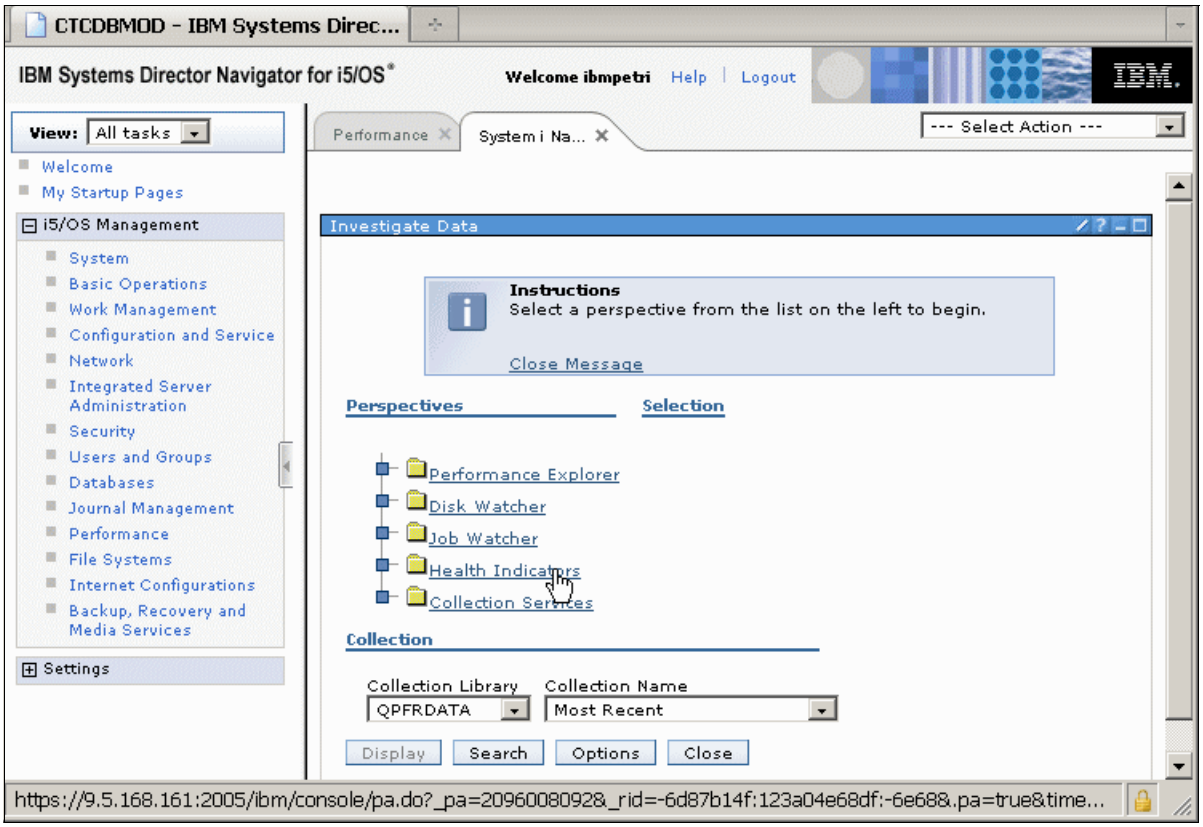


Figure 7-28 Selecting the Health Indicators

3. Select **Health Indicators**, which opens the window shown in Figure 7-29.

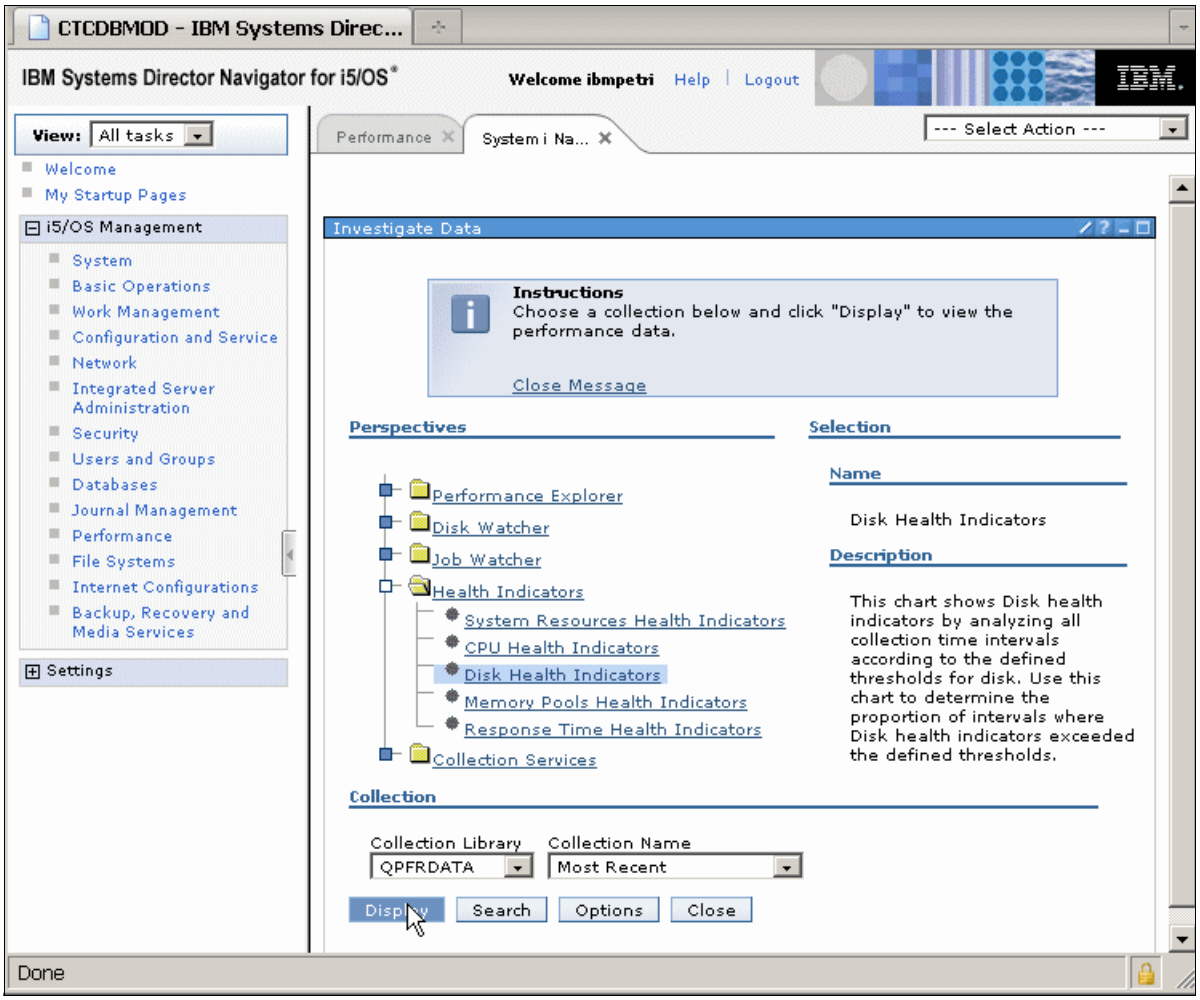


Figure 7-29 Select the Disk Health Indicators

4. Select **Disk Health Indicators** and click **Display**, which opens the window shown in Figure 7-30.

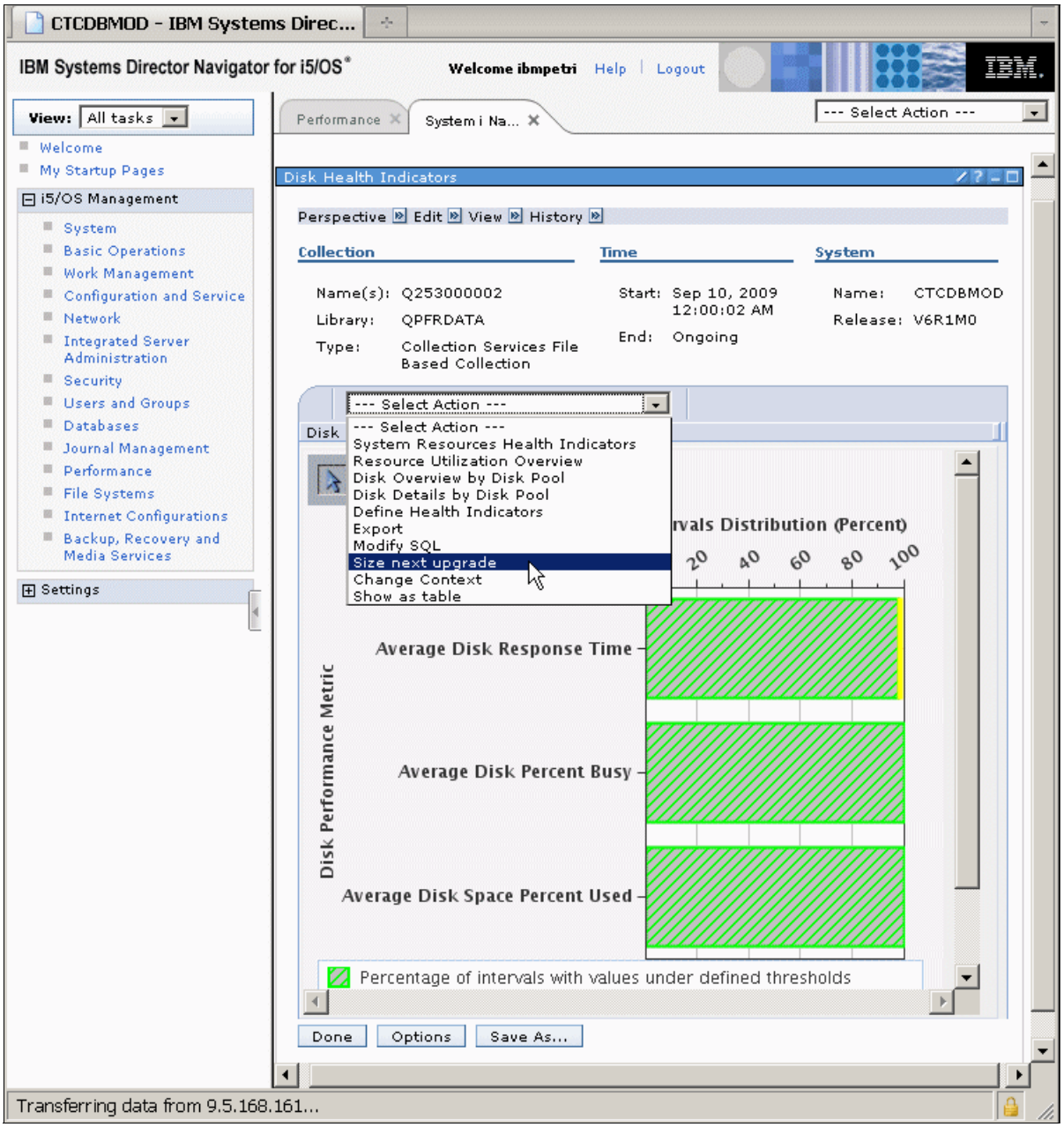


Figure 7-30 Select the Size next upgrade action

5. Select the **Select Action** drop-down menu and click **Size next upgrade**. The Size next upgrade window opens, where you may verify that you are sizing the correct partition, as shown in Figure 7-31.

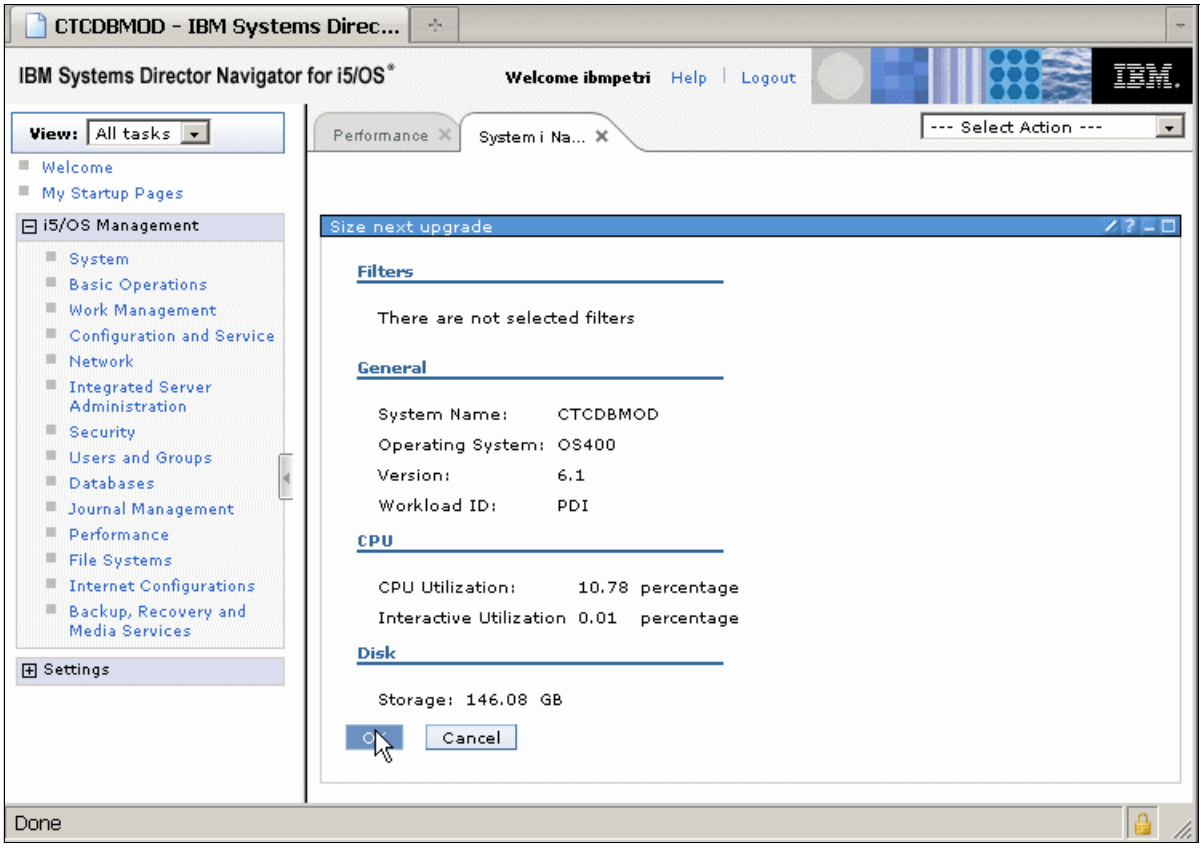


Figure 7-31 Verifying the partition to size

6. Click **OK** in the Security Warning window, as shown in Figure 7-32.

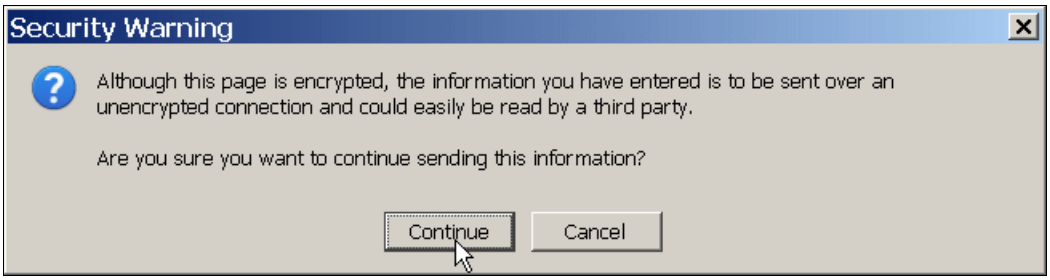


Figure 7-32 Security Warning

7. Click **Continue**. A new session window for the IBM Systems Workload Estimator Web site opens, as shown in Figure 7-33.

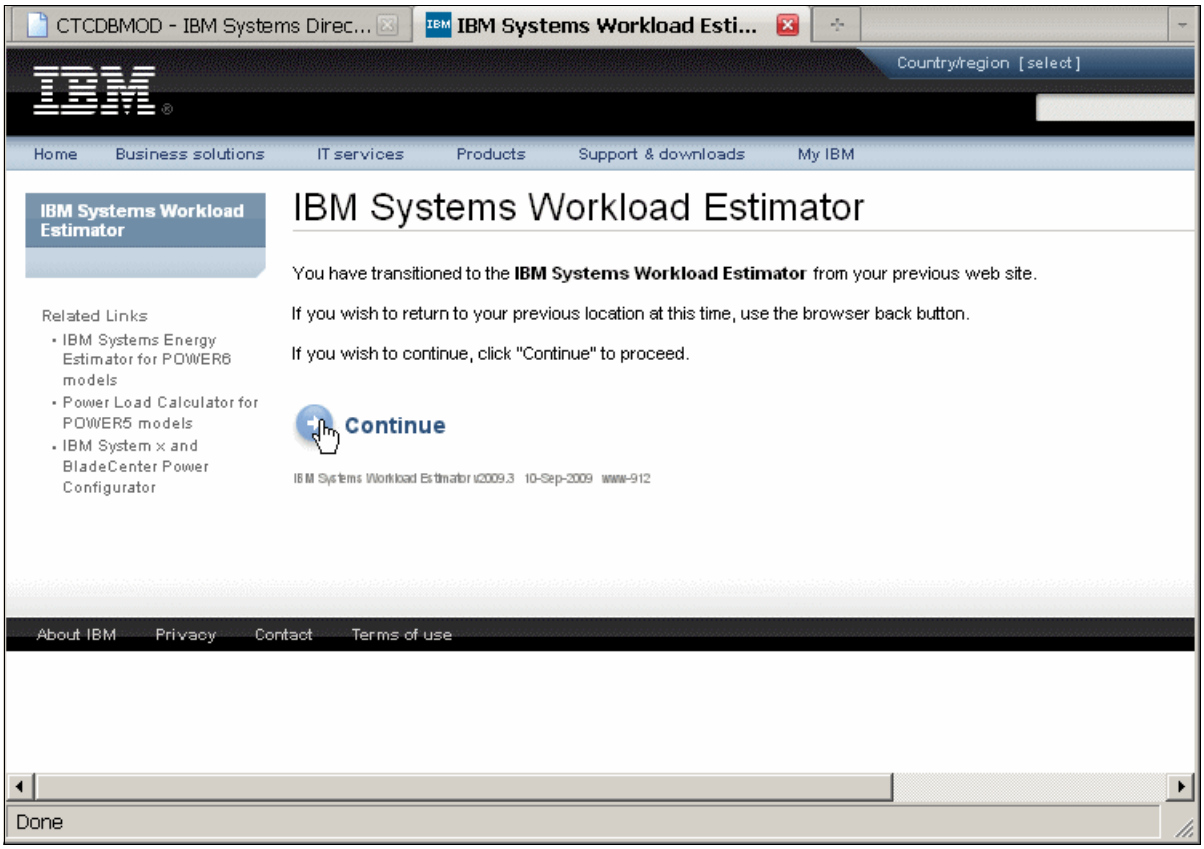


Figure 7-33 IBM Systems Workload Estimator tool

8. Click **Continue**. The PDI Workload Definition window opens, as shown in Figure 7-34.

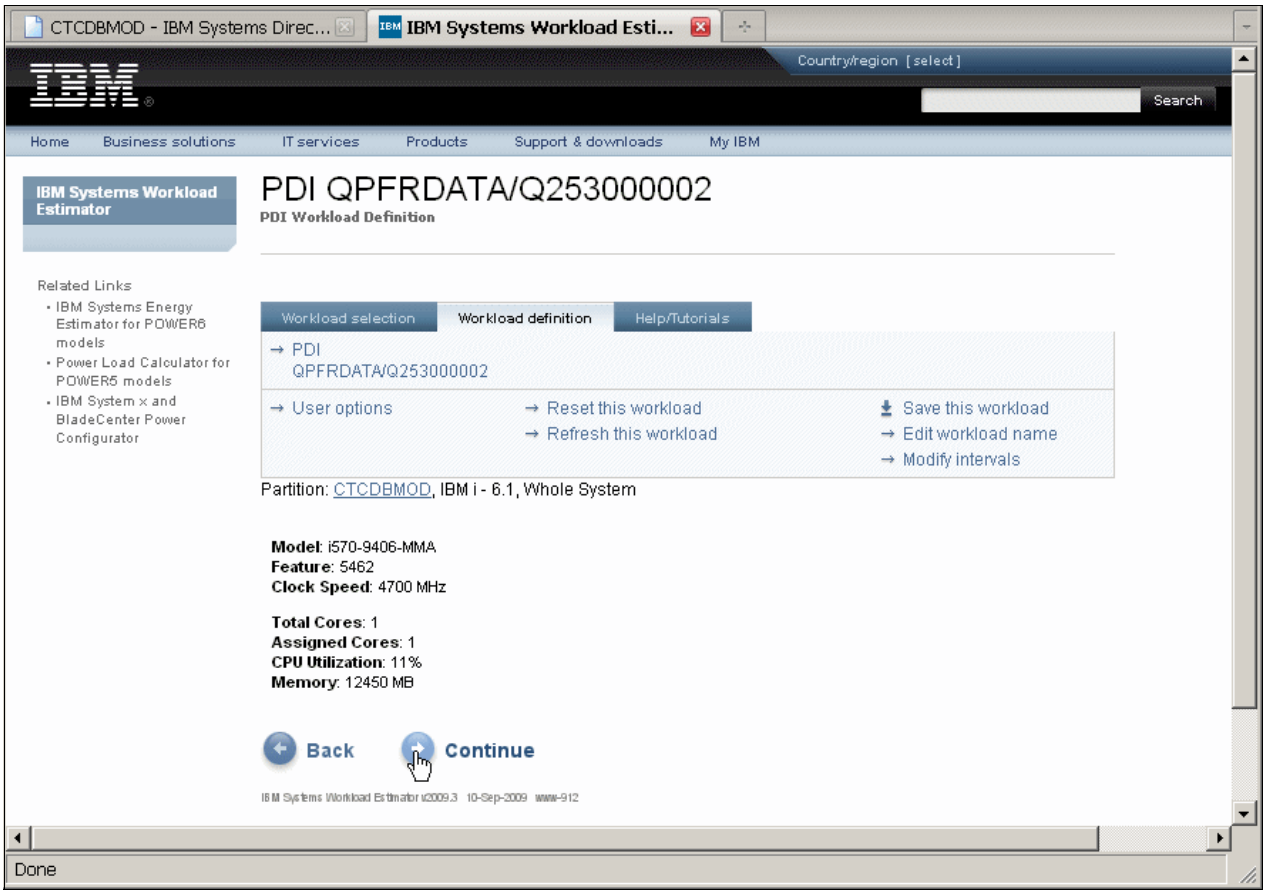


Figure 7-34 PDI Workload Definition window

Clicking **Continue**. The Selected system window opens, as shown in Figure 7-35.

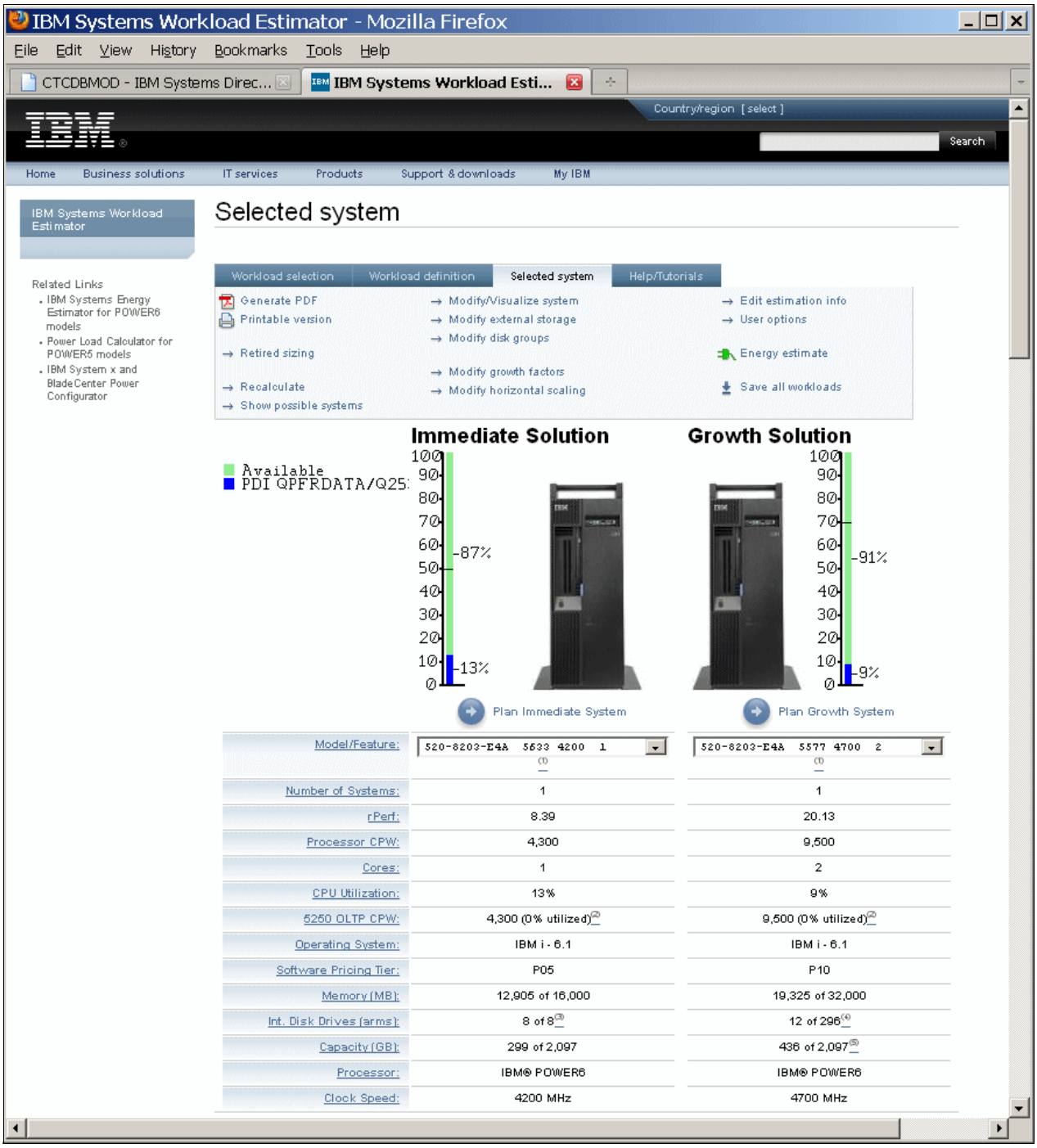


Figure 7-35 PDI: Selected system window

This window shows both the Immediate Solution and the Growth Solution physical systems that are capable of supporting the measured workload.

Note: Keep in mind that creating a subset of a collection in PDI to send to WLE has its advantages. If you use a perspective showing metrics over time, and select multiple intervals, the data that is sent to WLE will only represent the data between those intervals. This would be useful if you had run a workload on your current system between 2:00 and 4:00, and wanted to send the real performance metrics into WLE to use as a baseline for that workload. You could then project what double that workload might look like, for example.

Starting the Workload Estimator from scratch

If you are not using the PM for Power Systems service offering, using WLE is more difficult, as you must build and enter the workload manually.

Perform these steps:

1. Open a browser and enter the following URL to open the Workload Estimator:

   http://www-912.ibm.com/wle/EstimatorServlet

2. Click **New IBM Power System Estimation**. This changes the partition folder to the IBM i V6.1 system, as shown in Figure 7-36.

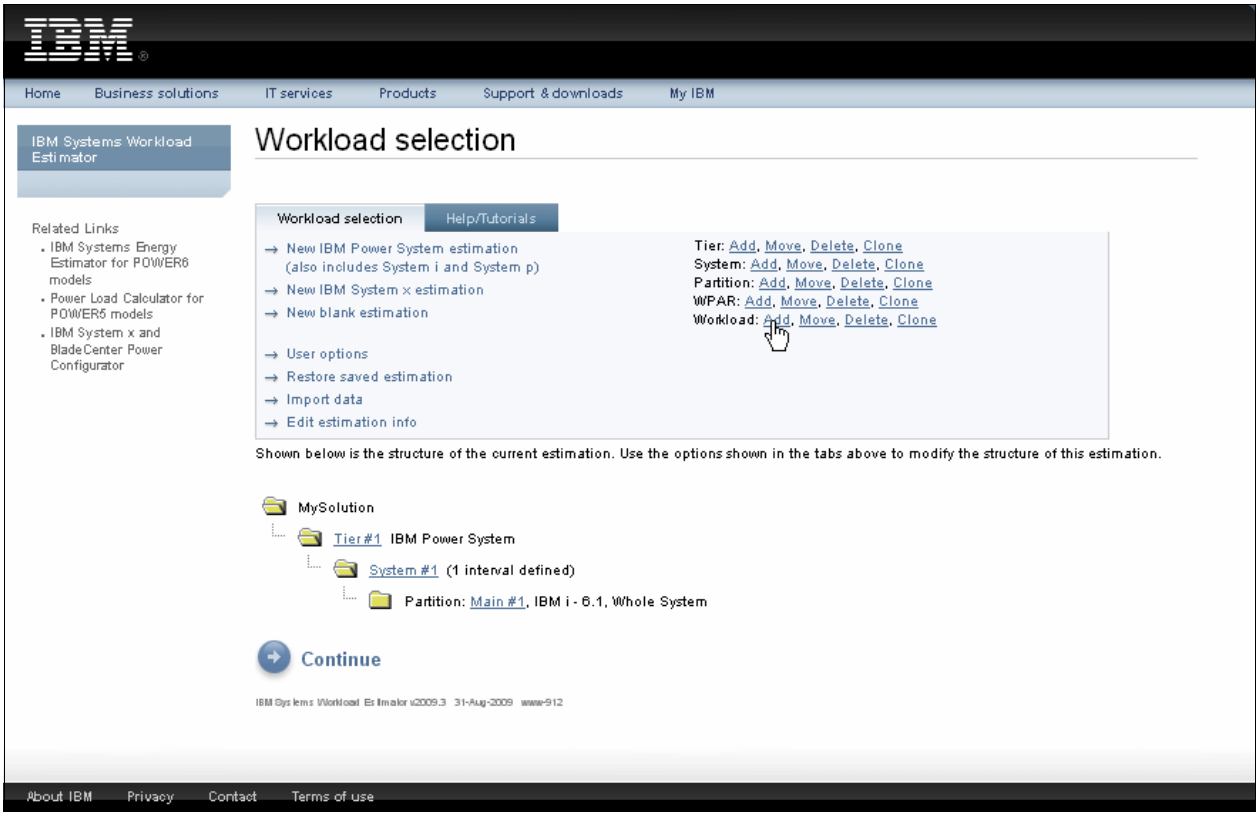


Figure 7-36 Sizing an IBM i system

3. Click the **Add Workload** link. The Workload selection window opens, as shown in Figure 7-37.

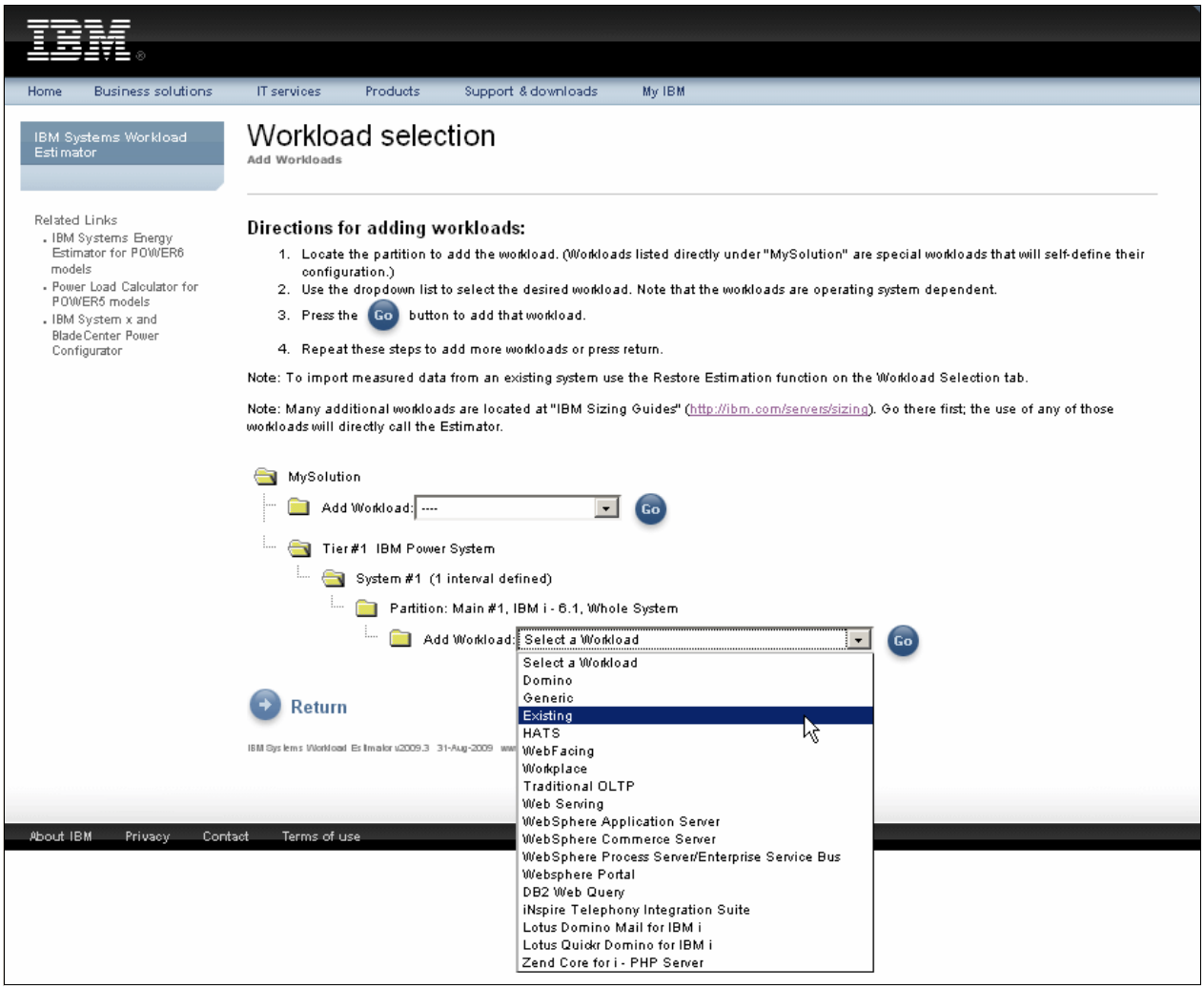


Figure 7-37 Select the existing workload

4. Open the Add Workload drop-down menu and select **Existing**. Click **Return**. The Workload selection opens, and you notice that is a little different: There is a yellow triangle next to the workload, as shown in Figure 7-38. That triangle means that the workload is not defined yet.

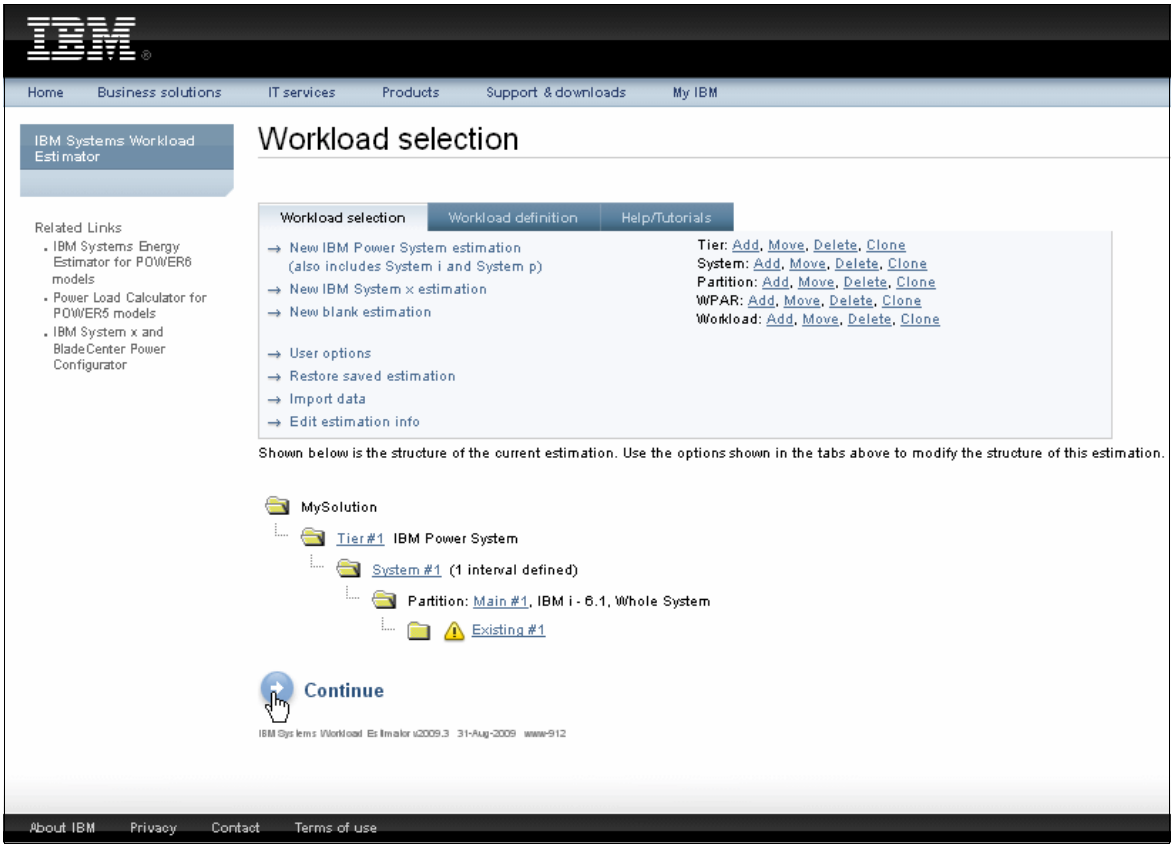


Figure 7-38 Defining the workload

5. Click **Continue** to start defining the workload. The first parameters (system type and the Operating System level) may be entered, as shown in Figure 7-39.

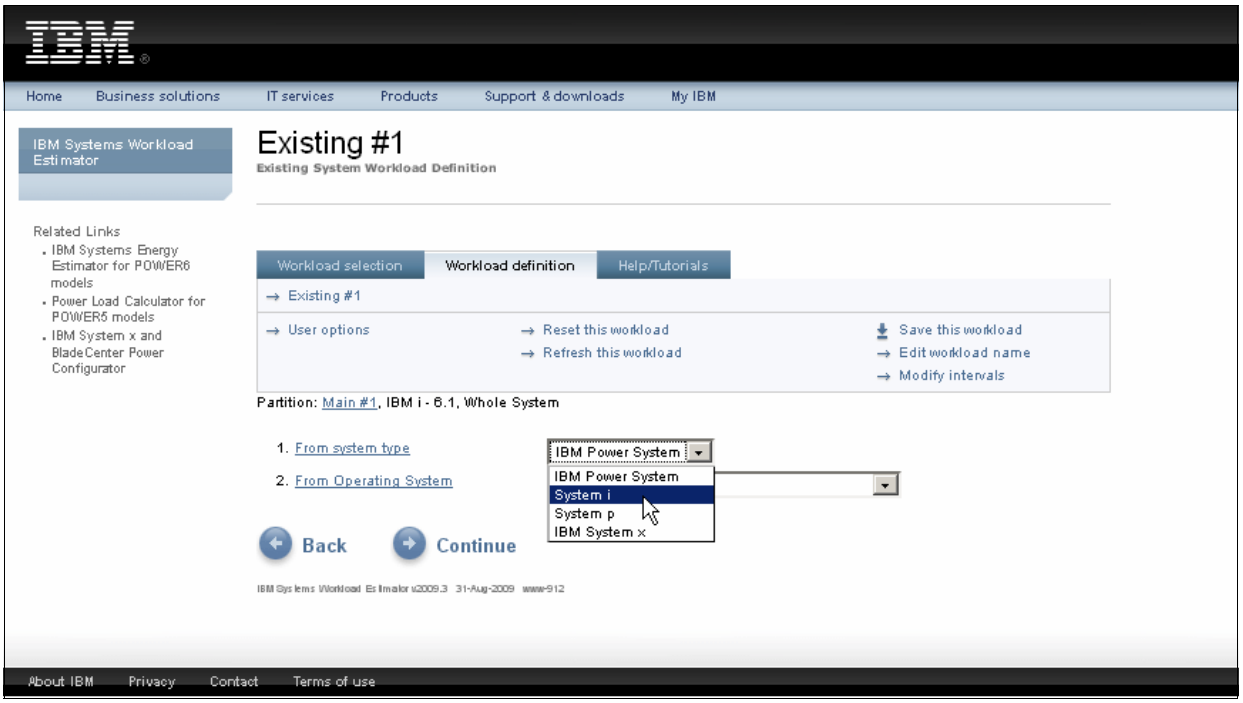


Figure 7-39 Defining the system to size

6. Select the appropriate system from the drop-down menu and select the operating system level, as shown in Figure 7-40.

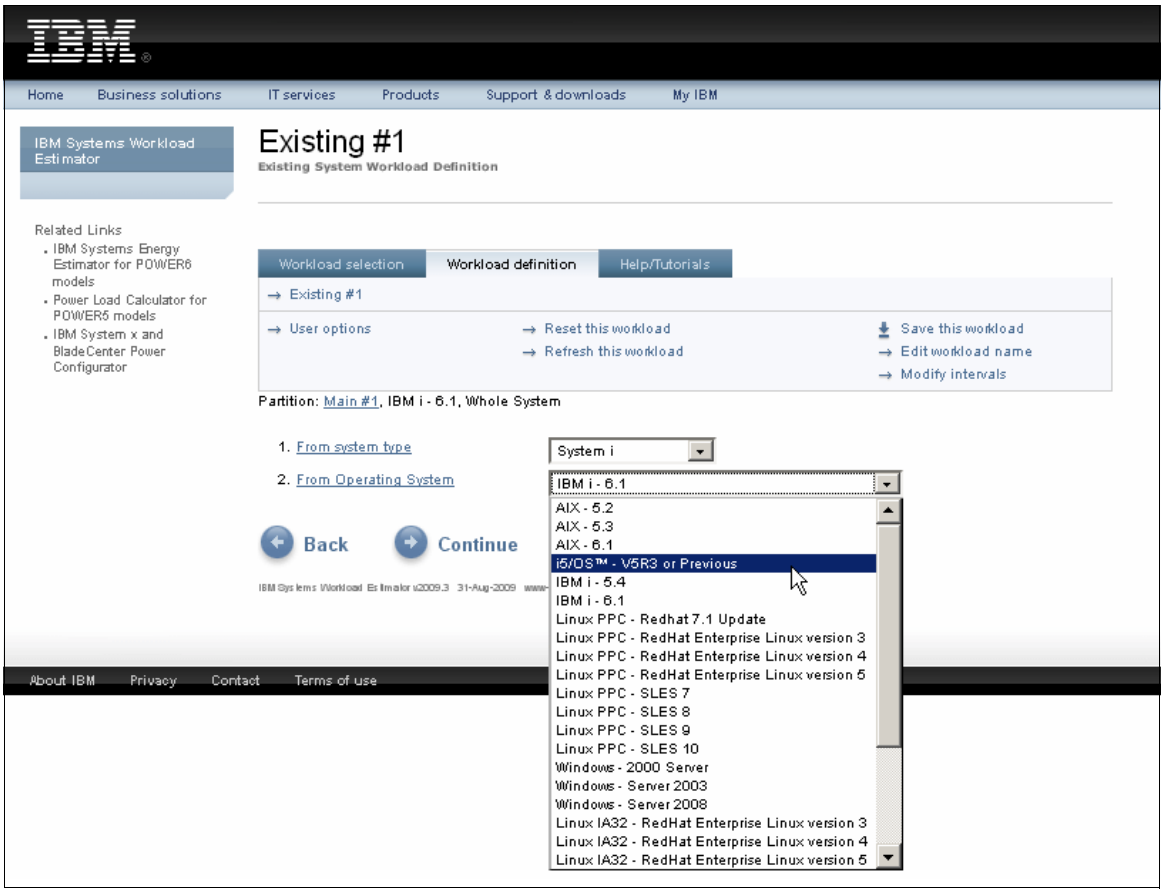


Figure 7-40 Select the operating system level

7. After defining the system type and the level of operating system, click **Continue**. The Workload definition window opens, as shown in Figure 7-41.

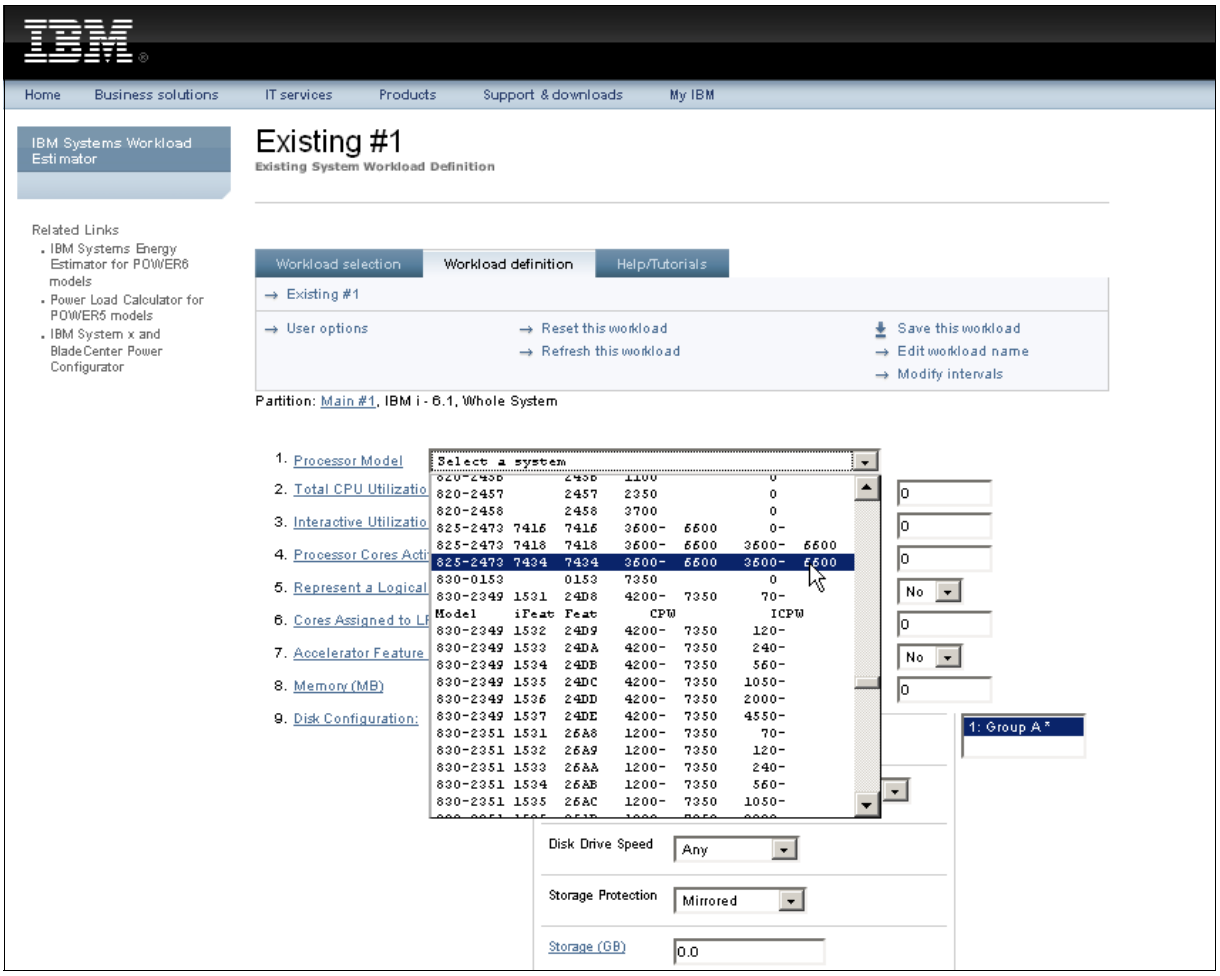


Figure 7-41 Select the processor model for the workload

8. Select the drop-down menu for the processor model. Scroll down the list until the correct one is shown and select it. Specify the values for the current CPU utilization and Interactive CPU utilization percentages and how many processor cores there are activated in the current hardware. You must specify the amount of memory, both of the Main Storage and the disk subsystem.

Specifying the disk subsystem characteristics is done in the same window, as shown in Figure 7-42.

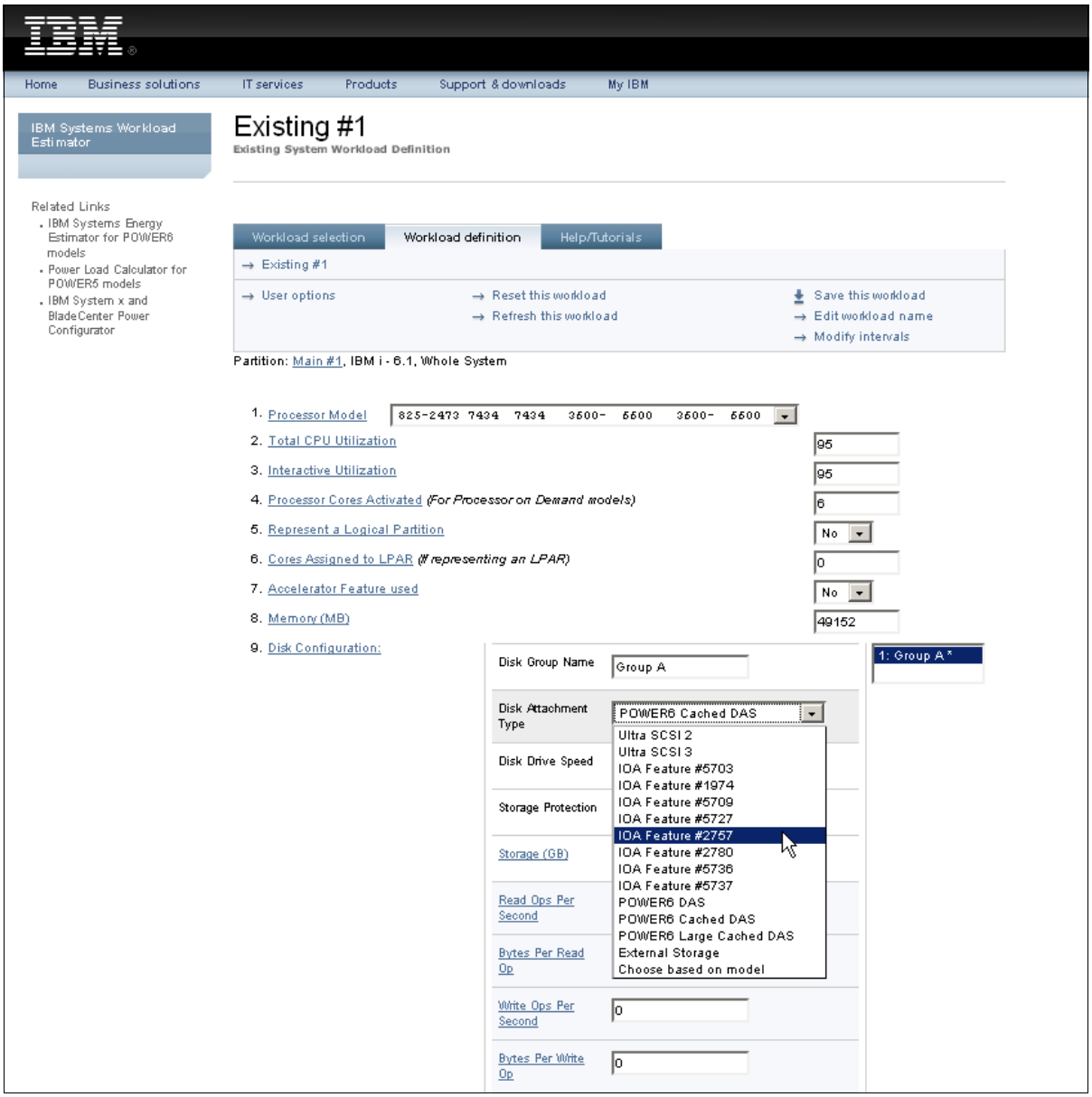


Figure 7-42 Specify the disk configuration

9. Select the drop-down menu and select the appropriate disk IOA type. Enter the disk drive speeds, the storage protection method, and the amount of disk space available.

> **Note:** You may either specify the number of disk operations and the size of disk operations per second or the number of disk arms and their percentage of busy.

In this example, we use the number of arms and the busy percentage, as shown in Figure 7-43.

IBM Systems Workload Estimator

Existing #1

Existing System Workload Definition

Related Links
- IBM Systems Energy Estimator for POWER6 models
- Power Load Calculator for POWER5 models
- IBM System x and BladeCenter Power Configurator

Workload selection | Workload definition | Help/Tutorials

→ Existing #1

→ User options
→ Reset this workload
→ Refresh this workload
Save this workload
→ Edit workload name
→ Modify intervals

Partition: Main #1, IBM i - 6.1, Whole System

1. Processor Model: 825-2473 7434 7434 3500- 5500 3500- 5500
2. Total CPU Utilization: 95
3. Interactive Utilization: 95
4. Processor Cores Activated *(For Processor on Demand models)*: 6
5. Represent a Logical Partition: No
6. Cores Assigned to LPAR *(If representing an LPAR)*: 0
7. Accelerator Feature used: No
8. Memory (MB): 49152
9. Disk Configuration:

1: Group A *

Disk Group Name	Group A
Disk Attachment Type	IOA Feature #2757
Disk Drive Speed	15,000 RPM
Storage Protection	Mirrored
Storage (GB)	750
Read Ops Per Second	0
Bytes Per Read Op	0
Write Ops Per Second	0
Bytes Per Write Op	0

Specify either ops and their sizes (the preferred method) or drives and busy percentage.

Number of Disk Drives	44
Disk Busy Percentage	20

Add new Group | Remove this Group

Figure 7-43 Specify the disk subsystem characteristics

10. The next parameters to enter are the protection method for the target system and the numbers for network operations, as shown in Figure 7-44.

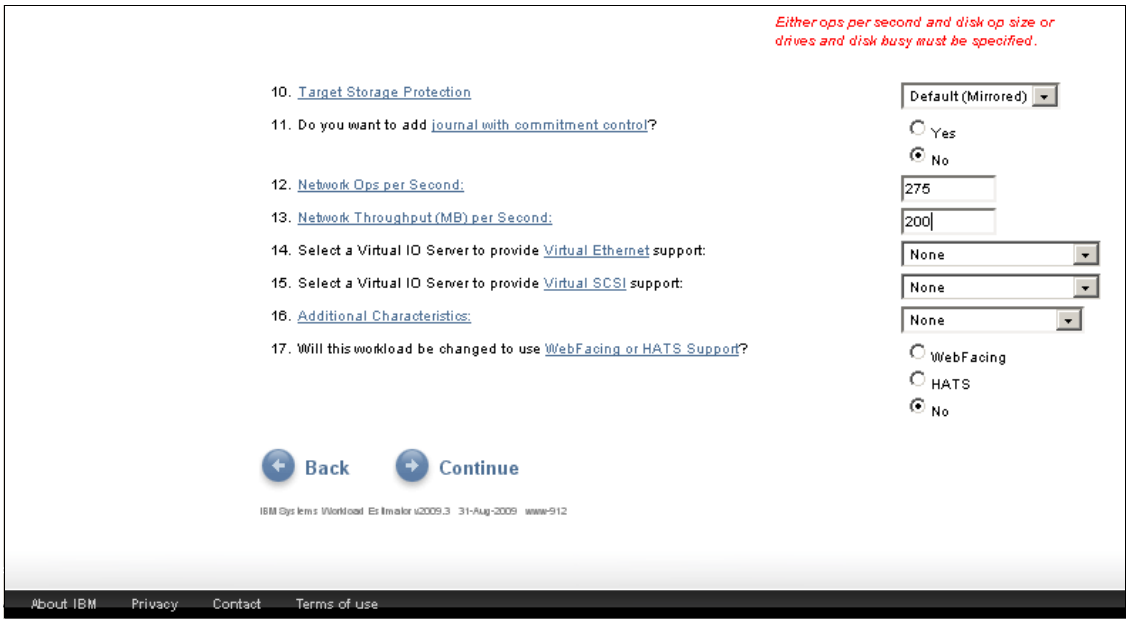


Figure 7-44 Specify the network load

11. Click **Continue** to select a system, as shown in Figure 7-45.

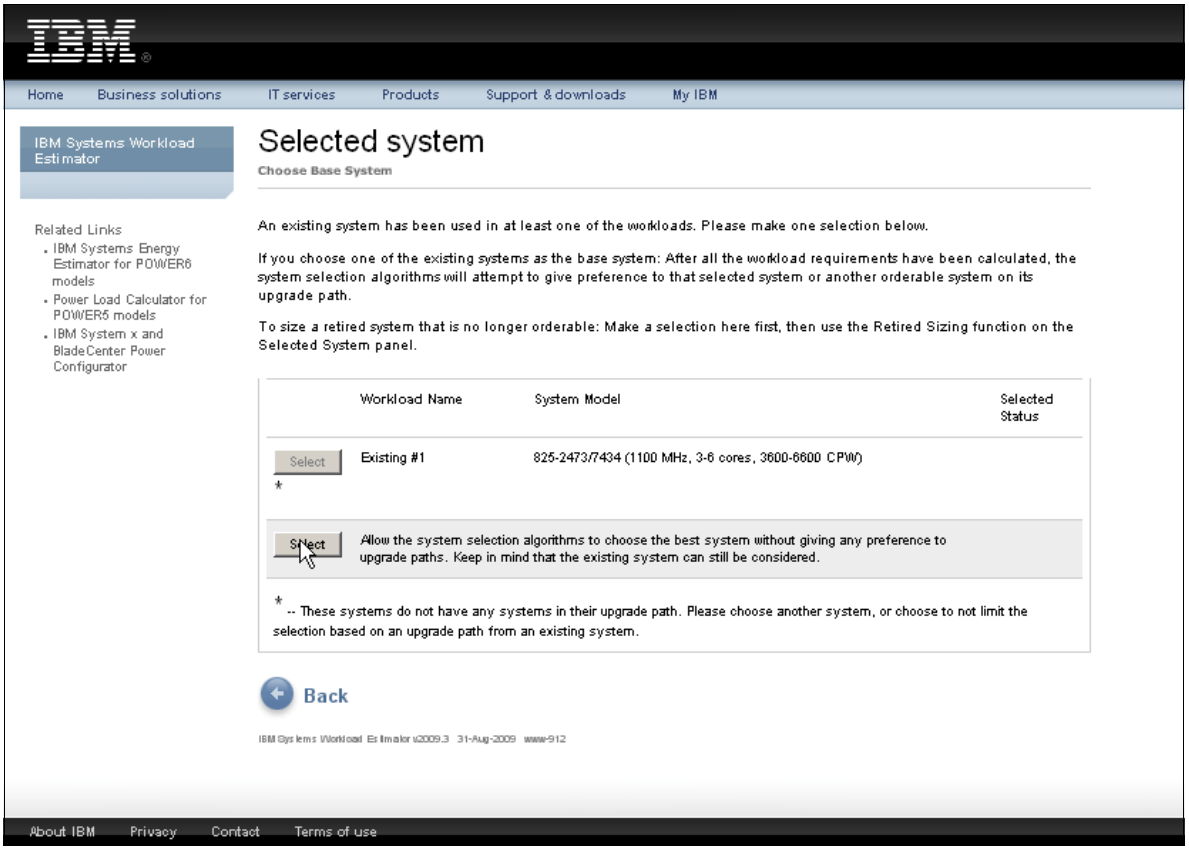


Figure 7-45 Selecting a base system for sizing

12. If there was a growth path available from the model 825 where our workload is coming from, we would have two systems from which to select. As there is only one path, clicking the lower button opens a window that shows both a system for the Immediate solution and one for a 50 percent growth in the workload, as shown in Figure 7-46.

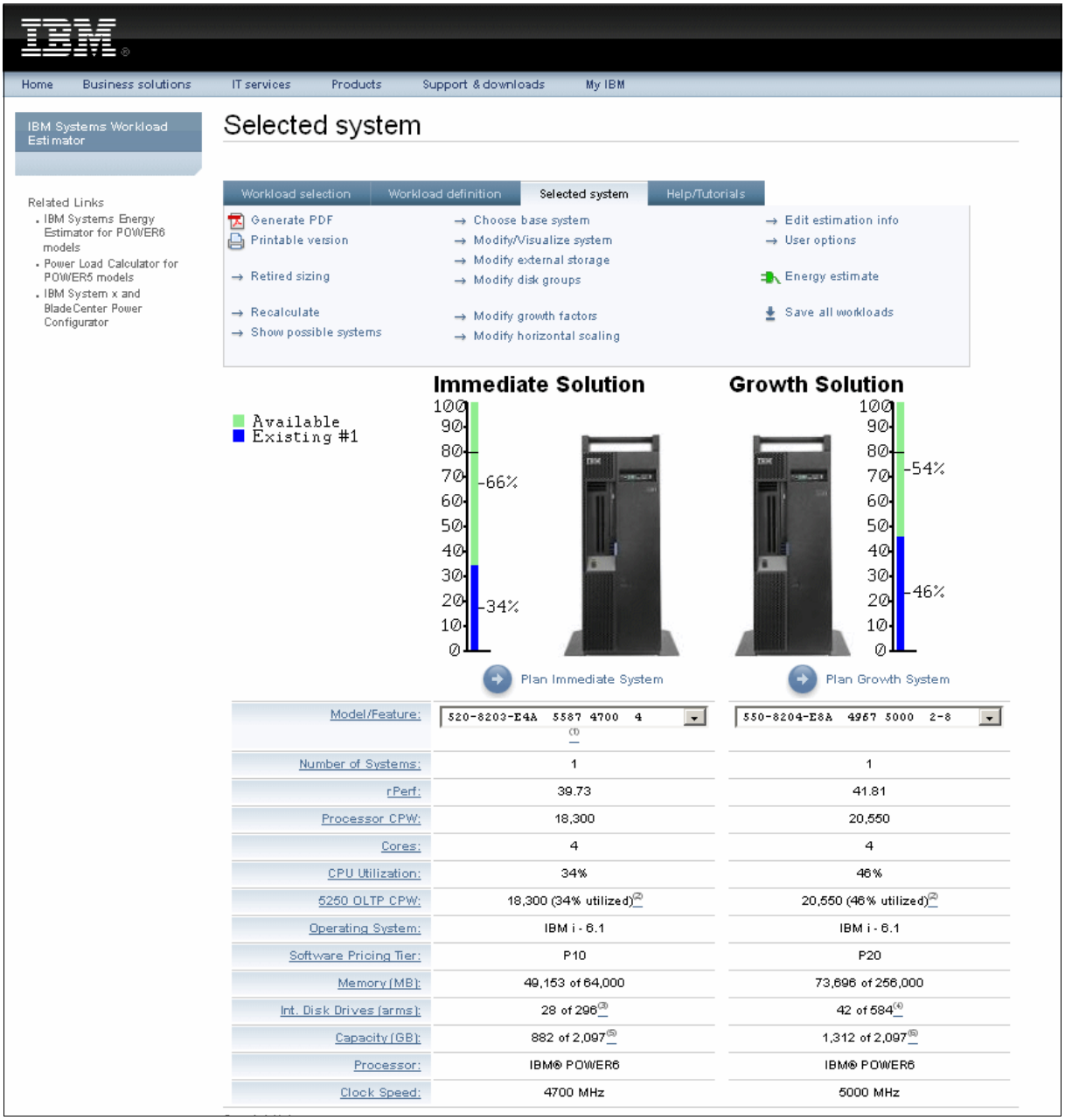


Figure 7-46 The Immediate and Growth Solutions

By default, this tool sizes for the latest hardware available. However, there is an option to size for earlier processor technology if you click the **Retired Sizing** link.

Even though you may use the stand-alone version of the Workload Estimator, we recommend that you use the version on the IBM Web site.

Part 3

Best practices and appendixes

In this part, we discuss some tuning tips and best practices. This part includes the following chapters and appendixes:

- Chapter 8, “Performance Management best practices” on page 215 discusses general tuning tips and system values.
- Appendix A, “Understanding disk performance metrics” on page 241 gives an explanation of disk performance metrics.
- Appendix B, Appendix B, “A partition as a business” on page 267 discusses work management terms.

© Copyright IBM Corp. 2009. All rights reserved.

8

Performance Management best practices

In this chapter, we discuss some tuning tips and best practices in Performance Management. The topics include:

- Back to the basics
- Resource utilization guidelines
- Key Performance Indicators
- Conducting a capacity planning cycle
- Effect of processor changes
- Tools for testing application performance
- Retaining collected data
- Disaster recovery and other failover systems
- Web application performance
- Performance Capabilities Reference

© Copyright IBM Corp. 2009. All rights reserved.

8.1 Back to the basics

This book contains information about some of the advanced features and technologies for managing performance on IBM i. Sometimes some of the basic tuning principles are overlooked, and doing so can lead to sub-optimal performance. In this section, we go over some of these basic tuning tips.

Note: Some of the tips in the following sections suggest changing values associated with memory pools, for example, activity levels and memory allocations. These changes are only applicable if you have the Automatically adjust memory pools and activity levels (QPFRADJ) system value set to 0. For more information about automatically adjusting memory pools and activity levels, refer to The Performance Adjuster (QPFRADJ) experience report in the IBM i Information Center by selecting **Related links** → **Experience reports** → **The Performance Adjuster (QPFRADJ)**.

We recommend setting QPFRADJ to ON for most installatioins. More experienced customers may want to set it to OFF so that they can have more control over the configuration.

8.1.1 System values

You can use IBM i performance system values to view and change priority, performance adjustments, and processing values for the system.

To access the performance category of system values from Navigator for i, select **Configuration and Service** and then select **System Values**.

Note: System values can also be found in the IBM Systems Director Navigator Web Console. They are only for that single partition.

Navigator for i uses descriptive names for system values. Table 8-1 provides a quick overview of the performance system values available in Navigator for i (and their counterparts in the character-based interface).

Table 8-1 Performance system values

Name in Navigator for i	Description of system value	Name in character-based interface
Base memory pool maximum eligible threads	This system value specifies the maximum number of eligible threads for the base memory pool.	QBASACTLVL
Base memory pool minimum size	This system value specifies the minimum base memory pool.	QBASPOOL
Communications arbiter jobs at restart	This system value specifies the number of communications arbiter system jobs that are available to process work for controllers and devices.	QCMNARB

Name in Navigator for i	Description of system value	Name in character-based interface
Communications configuration recovery	This system value specifies whether communications recovery attempts are made and how many attempts to make.	QCMNRCYLMT
Allow background database statistics collection	This system value specifies the types of requests for database file statistics collection that are allowed to be processed by system job QDBFSTCCOL.	QDBFSTCCOL
Dynamically adjust job priorities of interactive jobs	This system value sets the job priorities of interactive jobs.	QDYNPTYADJ
Dynamically adjust job priorities within priority bands	This system value sets the task scheduling policy.	QDYNPTYSCD
Lock libraries in a user job's library search list	This system value prevents other jobs from deleting or renaming the libraries in the search list.	QLIBLCKLVL
Maximum eligible threads	This system value specifies the maximum number of eligible threads.	QMAXACTLVL
Machine memory pool size	This system value specifies the size of the machine memory pool.	QMCHPOOL
Available display station pass-through server jobs	This system value specifies the number of pass-through server jobs that are available to process display station pass-through, iSeries Access workstation function (WSF), and other 5250 emulation programs on programmable workstations.	QPASTHRSVR
Automatically adjust memory pools and activity levels	This system value specifies when to automatically adjust memory pools and activity levels.	QPFRADJ
Processor multitasking	If the hardware on your system supports processor multitasking, this system value allows you to set the multitasking capability to be on, off, or System-controlled.	QPRCMLTTSK
Parallel processing for queries and indexes	This system value specifies whether to use parallel processing.	QQRYDEGREE
Database query time limit	This system value sets the time limit for a database query.	QQRYTIMLMT

Name in Navigator for i	Description of system value	Name in character-based interface
Automatically adjust thread resources	This system value specifies whether the system should dynamically make adjustments to the affinity of threads currently running on the system.	QTHDRSCADJ
Thread affinity	This system value specifies whether secondary threads have affinity to the same group of processors and memory as the initial thread.	QTHDRSCAFN
Move interactive jobs to base pool at end of time slice	This system value specifies whether to move interactive jobs to the base pool at the end of the time slice.	QTSEPOOL

Note: New System Values often appear in new versions or releases of IBM i. You should read the product documentation for each new version or release to ascertain whether new values have been created.

The system values provided in Table 8-1 on page 216 should be reviewed and set soon after the installation of IBM i and reviewed again after major operating system or hardware upgrades. The setting for each value will depend on the hardware configuration and the workloads being run, so it is not practical to provide recommended settings in this IBM Redbooks publication. However, the IBM i Information Center contains additional information that may assist you in determining appropriate vales for a given environment. The information can be accessed by selecting **Systems management** → **System values** in the Information Center.

8.1.2 Work with System Status tips

Work with System Status identifies page faulting and wait-to-ineligible transitions for each main storage pool.

Here are some useful tips:

- For interactive pools, typically you want the wait-to-ineligible transition values to be very small (less than 10% of the active-to-wait value).

Figure 8-1 shows where to find this value on the Work with System Status display. If you see any wait-to-ineligible value at all, increase the MAXACT value by 5 to 10 until the wait-to-ineligible is 0. Remember to press F10 to reset the statistics. Wait 10 seconds between refreshes.

```
                          Work with System Status                    CTCDBMOD
                                                            08/31/09  13:29:50
% CPU used . . . . . . . :         .9    Auxiliary storage:
% DB capability  . . . . :         .0      System ASP . . . . . . :     1411 G
Elapsed time . . . . . . :   00:00:19      % system ASP used  . . :     8.5959
Jobs in system . . . . . :        653      Total  . . . . . . . . :     1411 G
% perm addresses . . . . :       .007      Current unprotect used :     5765 M
% temp addresses . . . . :       .013      Maximum unprotect  . . :     6129 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   Active->  Wait->  Active->
 Pool   Size (M)  Size (M)  Active    Wait     Inel     Inel
   1      445.19    222.17   +++++    15.3       .0       .0
   2    10496.87      4.06     151   17627       .0       .0
   3     1215.80       .00     304    24.5       .0       .0
   4         .25       .00       5      .0       .0       .0
```

Figure 8-1 Work with System Status: wait-to-ineligible transitions

- The machine pool (pool 1) should have fewer than 10 faults per second, which is the sum of the database and non-database faults. You can ignore the Pages column. Figure 8-2 shows where to find this value on the Work with System Status display.

```
                          Work with System Status                    CTCDBMOD
                                                            08/31/09  13:46:35
% CPU used . . . . . . . :        2.0    Auxiliary storage:
% DB capability  . . . . :         .0      System ASP . . . . . . :     1411 G
Elapsed time . . . . . . :   00:01:30      % system ASP used  . . :     8.7818
Jobs in system . . . . . :        645      Total  . . . . . . . . :     1411 G
% perm addresses . . . . :       .007      Current unprotect used :     5383 M
% temp addresses . . . . :       .013      Maximum unprotect  . . :     6129 M

Type changes (if allowed), press Enter.

System    Pool    Reserved    Max   -----DB-----  ---Non-DB---
 Pool   Size (M)  Size (M)  Active  Fault  Pages  Fault  Pages
   1      463.42    223.28   +++++     .0     .0    5.5    6.0
   2     9870.73      3.99     151     .0     .0     .4     .7
   3     1823.71       .00     304    9.9   46.5    4.8   19.4
   4         .25       .00       5     .0     .0     .0     .0
```

Figure 8-2 Work with System Status: machine pool faults

- If only system jobs and subsystem monitors are running in *BASE, then the fault rate for that pool should be less than 30 faults per second.
- The basic method for tuning your storage pools is to move storage from pools with good performance to pools with bad performance. In this situation, you should measure performance in response time or as throughput. Continue to move storage until the pool with the bad performance gets better, or until the pool with the good performance gets worse. Do not decrease a pool by more than 10% at a time.

- Determining a good fault rate for the user pools can be complicated and will vary from pool to pool and from system to system. The fault rate alone is not necessarily a measure of good performance or bad performance.
  - For interactive jobs, look at how much the faults contribute to the end response time. This can be calculated by multiplying the faults per transaction by the disk response time.
  - For batch jobs, look at how much the faults add to the elapsed time of the job. This can be calculated by multiplying the total faults by the disk response time.
- Use the formula shown in Example 8-1 to calculate the approximate number of page faults per transaction.

Example 8-1 Calculating the approximate number of page faults per transaction

```
[(database faults + non-database faults) / active->wait] X 60
```

For more information about faulting rates in user pools, refer to “User pool faulting guidelines” on page 222.

8.1.3 Work with Disk Status tips

Work with Disk Status shows the percentage of space used and the percentage of time that the disk arms are busy. If the average percent for the Busy column is high (for guidelines, refer to 8.2.4, “Disk percent busy” on page 226), you may need to consider installing more disk arms. The suggestion to install more disk arms assumes that any page faulting problems have been addressed.

For more information about disk percentage busy, refer to 8.2.4, “Disk percent busy” on page 226.

8.1.4 Work with Active Jobs tips

The Work with System Activity display lists active jobs that have used any CPU in the last few seconds. The list is sorted in the order of the amount of CPU seconds that are used, which is the default view. If a high priority job (low number) is using a lot of CPU (>50%) for an extended period of time, then that job could cause the entire system to have poor response times. Here are some suggestions for improving your response time in this situation:

- If a job or a small set of jobs seems to be using a large percentage of the CPU, check the job priority (PTY). If the priority of the job is a lower number than the jobs with poor performance, you may want to consider changing the priority of the offending job or jobs. Use option 5 (Work with job), then option 40 (Change job), and specify a larger RUNPTY value (greater than the priority of the jobs that you want to ran faster).
- If the offending job is an interactive job that is running a job that is better suited to run in batch mode, you may want to contact the user and recommend one of the following:
  - That they submit their work as a batch job.
  - That they change the priority of the job to 50. 50 is the typical priority for a batch job.
- If the CPU utilization is high (>80%) and all jobs seem to have an equal, but small, CPU percentage, this situation could mean that you have too many active jobs on the system.

8.1.5 General tuning tips

Here are some general tuning tips:

- Favor output over input. On a system with constraints, it is usually better to throttle the number of new jobs or threads that are started, so that existing jobs or threads are not adversely impacted.
  - The activity level should not be too high. The term *too high* is relative to the number of jobs or threads expected to be processing in the memory pool at any point in time.
  - Allow the work to finish.
- Do not mix different types of jobs and priorities in the same pool. For example, do not mix class entries in subsystem descriptions.
- Remove batch jobs from *BASE by creating another batch pool. Route batch jobs to *SHRPOOL1. One batch job to a pool is ideal. This can be done by modifying the Job description (JOBD) for the routing data, and the subsystem description (SBSD) for the routing entry.
- On a shared production and development partition, remove programmers from the interactive pool (*INTERACT) by creating another interactive pool, *SHRPOOL2 (job description (JOBD) for routing data, subsystem description (SBSD) for routing entry).

8.2 Resource utilization guidelines

Performance management practices often include the concept of resource utilization guidelines or thresholds. Thresholds attempt to set a resource utilization rate as the point at which higher rates may produce unpredictable or degraded performance.

However, with IBM i and Power Systems, it is very difficult, if not impossible, to specify finite threshold values for hardware resources that will suit all environments. When asked questions such as *what is the recommended threshold for CPU utilization?*, the answer is typically *it depends*. This is due to the variability of the factors that are used to determine a threshold, such as processor speed, workload mix, and so on.

In this section, we attempt to address the lack of recommended guidelines by providing *rules of thumb*. Rules of thumb provide a starting point, from which you can fine-tune to determine the appropriate thresholds for your environment.

8.2.1 Memory

One relatively simple method to assess whether faulting rates may need further attention is to use the calculation shown in Example 8-2.

Example 8-2 Memory faulting eye catcher

```
(100 * (CPU utilization by the pool / 100)) * number of processing units
```

If the actual faulting rate is higher than the calculated rate, you may want to conduct some further analysis. A higher actual faulting rate does not necessarily mean it is causing degraded performance; it is simply an eye catcher to highlight the memory pool.

Consider the following example: Assume we have a system with 28 processing units and pool 5 is running at 36% CPU utilization. Using the eye catcher formula, the calculated fault rate is 1008 faults per second, as shown in Example 8-3.

Example 8-3 Memory faulting eye catcher example

```
(100 * (36 / 100)) * 28 = 1008
```

Note: The figures needed for the calculation are all contained in the Storage Pool Utilization report, which can be generate by running the PRTSYSRPT TYPE(*POOL) command.

If the actual faulting rate was much higher than 1,008, check to see what the Average Response time is for the pool. If it is acceptable, consider not taking further action on the faulting rate of this pool.

The IBM Technote “Memory Pool Tuning and Faulting Guidelines”[1] also contains some good tips. It has been reproduced below, with some minor updates.

This document highlights a couple of referenced procedures for tuning memory pools. This has been drawn from current and existing IBM online resources. While these are all publicly available, this document summarizes the key concepts.

Machine pool faulting guidelines

Use one of the performance data analysis tools to identify and correct the following possible problems:

1. If the page fault rate for the machine pool is higher than 10 faults/second, give the machine pool more memory until the fault rate falls below this level. If faults in the machine pool continue to be driven up and performance adjustment (QPFRADJ) is not turned on, the system may become completely unresponsive. This may include no messages being written to the history log and telnet sessions dropping.
2. If the disk utilization is greater than 40%, look at the waiting and service time to determine what portion the disk waiting and service time is of the average response time for transactions. If these values are unacceptable, you might need to reduce the workload to manage priorities.
3. If the Input/Output Processor (IOP) utilization is greater than 60%, consider adding an additional IOP and assign some disk resources to it.
4. If the page faults in the user pool are unacceptably high, refer to the IBM i Information Center topic “Automatically tune performance”, which can be found by selecting **Systems management** → **Performance** → **Managing system performance** → **Tuning performance** → **Automatically tune performance**.

User pool faulting guidelines

Due to the large range of Power Systems and preceding processors, and due to an ever increasing variance in the complexity of user applications, paging guidelines are no longer published. Only machine pool guidelines and system wide guidelines (the sum of faults in all the pools) are published. Even the system wide guidelines are just that, guidelines. Each customer must track response time, throughput, and CPU utilization against the faulting rates to determine a reasonable faulting rate.

[1] The original IBM Technote can be found at the following address:
http://www-01.ibm.com/support/docview.wss?uid=nas187a48f52002ed16b862571790050d098

The following are the choices for tuning user pools:

- Set the system value QPFRADJ to 2 or 3.
- Manually tune the user pools. Move storage around until the response times and throughputs are acceptable. The rest of this section deals with how to determine these acceptable levels.

To determine a reasonable level of faulting in user pools, determine how much the faulting is affecting the interactive response time or batch throughput. These calculations will show the percentage of time spent doing faulting. The following are some guidelines.

> **Note:** All of the data required for the following calculations can be printed with the PRTSYSRPT command.

The following sections assume that interactive jobs are running in their own pool and batch jobs are running in their own pool.

Interactive pool faults

flts	The sum of database and non-database faults per second during a meaningful sample interval for the interactive pool
rt	The interactive response time for that interval
diskRt	The average disk response time for that interval
tp	The interactive throughput for that interval in transactions per second (transactions per hour/3600 seconds per hour)
fltRtTran = the diskRt * flts / tp	The average faulting time per transaction
flt% = fltRtTran / rt * 100	The percentage of response time due to faulting

If flt% is less than 10% of the total response time, then there is not much potential benefit of adding storage to this interactive pool. However, if flt% is 25% or more of the total response time, then adding storage to the interactive pool can be beneficial.

> **Note:** It is very difficult to predict the improvement of adding storage to a pool, even if the potential gain calculated is high. There can be instances where adding storage might not improve anything because of the application design. For these circumstances, changes to the application design might be necessary.
>
> Also, these calculations are of limited value for pools that have expert cache turned on. Expert cache can reduce I/Os when given more main storage, but those I/Os might or might not be faults.
>
> These notes apply to both interactive and batch jobs.

Batch pool faults

flts	The sum of database and non-database faults per second during a meaningful sample interval for the batch pool.
flt% = flts * diskRt *100	The percentage of time spent faulting in the batch pool. If multiple batch jobs are running concurrently, you must divide flt% by the number of concurrently running batch jobs.

batchcpu% The batch CPU utilization for the sample interval. If higher priority jobs (other than the batch jobs in the pool you are analyzing) are consuming a high percentage of the processor time, then flt% will always be low. This means that adding storage will not help much, but only because most of the batch time is spent waiting for the processor.

To eliminate this factor, divide flt% by the some of flt% and batchcpu%. For example, newflt% = flt% / (flt% + batchcpu%). This is the percentage of time the job spent faulting compared to the time it spent at the processor.

Again, the potential gain of adding storage to the pool must be evaluated. If flt% is less than 10%, then the potential gain is low. If flt% is greater than 25%, then the potential gain is high enough to warrant moving main storage into this batch pool.

DB2 for i memory sharing considerations

The discussion in this section was taken from the "IBM i V6.1 Performance Capabilities Reference"[2]:

DB2 for i has internal algorithms to automatically manage and share memory among jobs. This eliminates the complexity of setting and tuning many parameters which are essential to getting good performance on other database products. The memory sharing algorithms within SQL Query Engine (SQE) and IBM i will limit the amount of memory available to execute an SQL query to a "job share". The optimizer will choose an access plan that is optimal for the job's share of the memory pool and the query engine will limit the amount of data it brings into and keeps in memory to a job's share of memory. The amount of memory available to each job is inversely proportional to the number of active jobs in a memory pool.

The memory-sharing algorithms discussed above provide balanced performance for all the jobs running in a memory pool. Running short transactional queries in the same memory pool as long running, data intensive queries is acceptable. However, if it is desirable to get maximum performance for long-running, data-intensive queries, it may be beneficial to run these types of queries in a memory pool dedicated to this type of workload. Executing long-running, data-intensive queries in the same memory pool with a large volume of short transactional queries will limit the amount of memory available for execution of the long-running query. The plan choice and engine execution of the long-running query will be tuned to run in the amount of memory comparable to that available to the jobs running the short transactional queries. In many cases, data-intensive, long-running queries will get improved performance with larger amounts of memory. With more memory available, the optimizer is able to consider access plans that may use more memory, but will minimize run time. The query engine will also be able to take advantage of additional memory by keeping more data in memory, potentially eliminating a large number of DASD I/Os. Also, for a job executing long-running performance critical queries in a separate pool, it may be beneficial to set QQRYDEGREE=*MAX. This will allow all memory in the pool to be used by the job to process a query. Thus, running the longer-running, data intensive queries in a separate pool may dramatically reduce query run time.

8.2.2 Disk storage

Here we have a guideline for disk storage.

[2] This article can be found at the address http://www-03.ibm.com/systems/resources/pcrm_oct2008.pdf and is reproduced in Appendix A, "Understanding disk performance metrics" on page 241.

Problem summary

What is the guideline value for Auxiliary Storage Pool (ASP) percentage on WRKSYSSTS?

Resolution

The guideline value for acceptable performance is 80%. This is only a guideline, but around this point disk fragmentation starts to have a greater affect on performance. There is a threshold value for the ASP that is set in the service tools (SST or DST) that warns the user by way of a message that the ASP(s) has reached a specified percentage full.

8.2.3 Disk response time

Disk response times can vary significantly and still provide acceptable response times. Alexei Pytel of IBM Rochester authored three excellent “Understanding Disk Performance Metrics” articles. In part, he suggests:

> “The popular question is - what is the recommended guideline for *good* disk response time. There is no easy answer to that. There are too many factors, which impact this - disk technology, cache friendliness of a disk workload, mixture of reads and writes, tolerance of the particular application to the disk delays.
>
> As a rule of thumb, with the current disk technology, I would say that average response time less than 5 milliseconds is good, between 5 and 10 milliseconds is normal, above 10 milliseconds requires analysis and above 100 milliseconds is bad.”

For a full copy of Alexei’s articles, refer to Appendix A, “Understanding disk performance metrics” on page 241.

An alternative to grouping disk response times into categories is to establish disk response time objectives for a given application. This objective then becomes the benchmark for normal disk response time. If disk response times are below the benchmark, they can be considered good. If they exceed the benchmark, they can be considered as needing analysis.

For example, let us assume that the transaction response time objective for a given Online Transaction Processing (OLTP) application is 1 second. Also assume that the CPU seconds per transaction is 0.25 seconds. That implies that the disk response time for each transaction must be less than 0.75 seconds (allowing for other transaction components such as queuing for the CPU). In our example, assume we have an average of 50 synchronous disk I/Os per transaction. Therefore, the disk response time objective can be calculated by dividing the total disk response time per transaction by the average number of synchronous disk I/Os per transaction, as shown in Example 8-4.

Example 8-4 Calculating disk response time objectives: OLTP

```
0.75/50 = 0.015 seconds (or 15 milleseconds)
```

In this example, a disk response time of around 15 milliseconds can be considered normal. Below 15 milliseconds is good. Disk response times above 15 milliseconds may require further analysis to determine if they can be improved.

Batch processing is more susceptible than OLTP processing to longer disk response times. Typically, batch processing is significantly more disk I/O intensive.

Consider the following example: A business has a requirement that a particular batch process must complete in under 1,000 seconds (just over 15 minutes). Similar to our previous example, let us assume the CPU component of the job is 600 seconds, leaving less than 400 seconds for all disk I/O. Again, also assume that the batch process averages 100,000 synchronous disk I/Os each time it executes. To calculate the disk response time objective, we can use the same calculation as the previous example, as shown in Example 8-5.

Example 8-5 Calculating disk response time objectives - batch

```
400/100,000 = 0.004 seconds (or 4 milleseconds)
```

In this example, a disk response time around 4 milliseconds can be considered normal. Below 4 milliseconds is good. Disk response times above 4 milliseconds may require further analysis to determine if they can be improved.

8.2.4 Disk percent busy

Again, we refer to Alexei Pytel's "Understanding Disk Performance Metrics" articles:

> "This is probably the most often used measure of the health of a disk subsystem. This metric shows a percentage of the time disk was busy processing disk operations. This metric is also often - incorrectly - called disk utilization. Disk percent busy and disk utilization were the same thing when disk units were capable of processing only one operation at a time. Now the execution of two or more disk operations can overlap. Technically, if disk is 100% busy, it may still be able to perform additional disk operations - in addition to those which are already in progress. So we cannot really say that disk utilization is 100% in this case.
>
> Which value of this metric is considered normal? All performance guidelines are to a certain extent arbitrary. Years ago the rule used to be to keep disk percent busy below 40%. Benchmarks show, that with the faster disk hardware and with ability of new storage adapters to process more than one operation in parallel, acceptable response times can be achieved at up to 70%. IBM Workload Estimator uses significantly more conservative guideline of 25%. This number is heavily biased towards OLTP environments, which are more sensitive to variations in disk response time. It also makes room for a natural workload growth. Batch-oriented workloads may be able to tolerate much higher disk utilizations than that."

Batch-oriented workloads may be more tolerant because in many cases, business operations do not worry about how long a batch process takes, as long as it is completed by a set time. For example, many applications have an overnight processing component, where the resulting output is required for the business users the following day. In this scenario, the business user is not concerned whether the jobs end at 1:00 a.m. or 5:00 a.m., as long as they are finished by, say, 7:00 a.m.

In other situations, batch processing can be very sensitive to variations in disk percent busy. Consider a warehouse application where a user selects an option to print a packing slip for a delivery. The application is designed to submit the print job to batch. In this situation, the user may not be able to proceed with any more manual packing work until the packing slip is ready. Therefore, it is time sensitive. High disk percent busy metrics may slow the printing process.

If a system is exhibiting high disk utilization, it may be more prudent to inspect the disk response times. Disk response times provide a better indication of impacts to application performance.

8.3 Key Performance Indicators

Performance indicators are measurements that can be used to gauge how well a system is performing over time. The IBM i performance data collectors gather thousands of metrics that can be used as performance indicators. You should select a small number metrics that make sense to your business and are easily measured. These measurements are known as Key Performance Indicators (KPIs). There is little value in reporting on KPIs that are hard to explain to the intended audience, or metrics that take a significant amount of time to gather or calculate.

KPIs can be grouped into different categories. For IBM i, two suggested categories are:

- Hardware resource utilization
- Throughput

Tracking KPIs in both categories can avoid situations such as where the hardware is performing at levels below thresholds but the throughput is not meeting business expectations. It also helps to account for environmental factors such as seasonal variations and the economic climate. For example, if hardware resource utilization KPIs and throughput KPIs are increasing or decreasing at a similar rate, that may indicate a change in business volumes caused by seasonal change or economic upturn or downturn. However, if they are varying at significantly different rates, that may indicate changes unrelated to external influences.

The following sections describe the KPI categories in more detail and suggest some KPIs that may be suitable. A business may decide to define additional KPIs that have particular meaning to that business.

8.3.1 Hardware resource utilization

Hardware resource utilization needs little explanation. It is simply measuring how much of a resource is being used. Some suggested KPIs are:

- Total CPU utilization
- CPU usage by job priority:
  - Interactive
  - Batch
- Memory faults per second by memory pool:
  - Database
  - Non-database
- Disk:
  - Arm utilization
  - Response time
  - Wait time
  - Space utilization

These KPIs can then be tracked over time to determine if they are relatively static, increasing, or decreasing. They can also be tracked against predefined thresholds and if the measurements approach or reach thresholds, appropriate action can be taken.

8.3.2 Throughput

Throughput KPIs are often more meaningful to a non-technical audience. They are metrics that can describe the performance of a system in almost business terms. In a perfect world, it would be advantageous to report performance in true business terms, for example:

- How long it takes to enter a new contract.
- How long it takes to prepare an invoice.
- How many Electronic Fund Transfer (EFT) transactions processed per hour, and so on.

IBM i does not have any built-in mechanism to capture these metrics because it does not know what constitutes a business transaction. However, if an application has been designed and built to capture these metrics, and externalizes them to a data collector such as Collection Services, reporting is possible on true business transactions.

In the absence of true business metrics, IBM i provides metrics about the components that typically make up a business transaction. For example, in an application that has an *enter new contract* function, the function may be a series of five interactions with a user. These may be five 5250 terminal emulator transactions or five Web page transactions. Tracking the metrics of these transactions provide data that has more meaning to business users. Some suggested KPIs are:

- OLTP transactions (5250 or HTTP) per hour:
  - Peak
  - Average
- Average response time per transaction
- CPU usage per transaction
- Disk I/O per transaction
- Elapsed time of batch jobs

8.4 Conducting a capacity planning cycle

This section discusses the tasks that are typically undertaken during a capacity planning cycle.

8.4.1 Review current performance

The purpose of reviewing the current performance is to identify any resource bottlenecks. These bottlenecks may skew capacity planning predictions. Bottlenecks should be eliminated where possible and, if any remain, plan to make allowances for them when predicting future requirements.

Initially, the system should be checked to confirm that the current KPI values are within the current target KPI values. Particular attention should be paid to the current hardware resource utilization KPIs. If these are above target, it may impact predictions.

8.4.2 Select representative data

Representative data is needed as a base for a capacity plan. If incorrect data is selected, the capacity plan may include under or oversized predictions of future infrastructure requirements. When selecting representative data, consider the workload (using throughput volumes and response/run times and other KPIs). Compare these factors in multiple data collections to verify that the selected data is representative of normal business operations.

Also consider the variability of performance data when it is influenced by seasonal or economic conditions.

Another consideration for selecting representative data is to identify peaks within short collection periods. For example, once seasonal peaks have been identified, identify any processing peaks, which are usually triggered by business workflow practices, that occur within the season peaks. These *peaks within peaks* should be allowed for in the estimation phase, to ensure an acceptable throughput is achieved during these peaks in the future.

Once representative data has been selected, it should be reviewed to eliminate *exceptional conditions* within the data. A example of an exceptional condition may be where a job has consumed excessive resources because of an unusual error condition, such as looping. If this was evident throughout a data collection, consider using a different collection. If the excessive resource consumption was only for a short period during the collection, consider excluding either the job or the intervals during which it occurred.

8.4.3 Estimation

The estimation phase predicts two things: the likely value of KPIs and the underlying infrastructure required to support those values, to ensure they are within future target KPI values. Selecting an appropriate tool, such as the Workload Estimator, will automate much of the estimation process, but the key is to gather realistic projections of the business factors that impact performance. The focus should be on business factors that will likely cause changes in workloads on a system. Consider the following growth or declining factors:

- Number of transactions
- Number of users
- Longevity of current applications
- New functions within existing applications
- New applications
- Interfaces with other system

The other important data that needs to be gathered is the business' target throughput KPI values. These may or may not be the same as the current KPI values. In most cases, a business would want to see values maintained or improved. However, in some cases, a business may be prepared to lower their performance expectations if it affords costs savings in the future. Care should be taken when this approach is used; it will likely lead to user frustration in the future and require additional infrastructure to rectify it.

A key consideration when estimating future requirements can be accounting for *pent up demand*. In a currently constrained system, a demand may exist for system resources during periods of the day but the system does not have the capacity to service the demands. When this occurs, businesses often adjust their workflow to accommodate the lack of resources (for example, run additional or extended shifts). Typically, when resources become available (for example, a hardware upgrade or significant performance improvements in a new operating system release), the business reverts to their normal business practices. When sizing future infrastructure requirements, determine whether any pent up demand exists and, if it does, attempt to quantify it. Unfortunately, there are no tools available than can accurately measure this type of pent up demand. It may require the use of business process reviews to identify and quantify it.

8.5 Effect of processor changes

When contemplating changes to the processor configuration in a system, be aware of the differences between processor speed and processor capacity. Commercial Processing Workload (CPW) is a relative measure of performance of systems running the IBM i operating system and is sometimes, incorrectly, used a gauge of the speed of a system. CPW measures throughput, not speed.

For example, assume a system has a single processor running a batch process that is comprised of 50% CPU time and 50% disk response time. If the processor speed of the system is doubled (typically by *replacing* the processor), you could expect to see a significant reduction in the CPU component.

If, however, the total CPW rating of the system was increased by *adding* more processors, it is unlikely to improve the run time of the job. In this scenario, the system now has the capacity to run additional workload, but it will not reduce the original job's run time.

For more information about CPW, refer to the *Ideas International Evaluation of IBM Commercial Processing Workload (CPW) for Rating Relative Performance of IBM eServer iSeries Systems* white paper, which can be found at:

`ftp://ftp.software.ibm.com/systems/i/perfmgmt/iicpw.pdf`

8.6 Tools for testing application performance

For environments where performance is critical, consider using testing tools that can be used to test the performance of an application, before and after a major change. Major changes include:

- In house developed application releases
- Independent Software Vendor (ISV) application releases
- Hardware configuration changes
- Network configuration changes
- Upgrades to operating systems

These tools typically have the ability to record and playback test scripts and record the results.

Before a major change is implemented, run a set of test scripts to record the performance of the system prior to the change. Then, after the change, run the same scripts and compare the results. If any anomalies are found, determine whether they were caused by the change.

Where possible, run these tests on a test system before implementing the same change on a production system. Obviously, this may not be applicable for hardware changes if the changes are only being implemented on a single partition production system.

The IBM Rational® Performance Tester is one such tool that can be used to conduct automated performance regression testing on IBM i applications. For an example about how this tool can be used, refer to the IBM Redpapers™ publication *An Experience Using Rational PerformanceTester to Benchmark Oracle EnterpriseOne*, REDP-4456, which can be found at the following address:

`http://www.redbooks.ibm.com/abstracts/redp4456.html`

For more information about the product, refer to the Rational Performance Tester Web page found at the following address:

http://www.ibm.com/software/awdtools/tester/performance/

8.7 Retaining collected data

As mentioned in 4.2.4, “Managing Collection Services data” on page 58, Collection Services includes a function to automatically delete collected data after a specified interval. While this is a very useful feature to minimize disk storage utilization, it also limits the ability to compare current performance data against historical performance data. This ability is key when answering one of the first questions after a performance problem is reported: What has changed?

If you have data from older periods where performance was meeting expectations, you can compare the old and new data to see which KPIs have changed. This can often be a good starting point for an investigation because it narrows down the starting point.

The question then becomes, what data should I keep and for how long? As with most performance analysis related question, the answer is, it depends. This section contains some suggestions. These suggestions apply equally to all collected data, not just Collection Services, such as Job Watcher and Disk Watcher.

8.7.1 Store data centrally

If you have multiple physical or logical partitions, consider housing all collected data on a single partition. Where possible, this should not be a partition running production business operations. If the data being analyzed using the Investigate Data functions is large, it can be resource intensive.

Storing data centrally also simplifies backup and archival processes.

8.7.2 Keep some data online

It can be useful to retain about 10 days worth of data on disk, storage space permitting. This allows you to respond quickly to reports of poor or degrading performance. The data can be retained only on the partition it was created on or on a centralized partition.

8.7.3 Saving data to offline media

In your backup strategy, consider treating your performance data the same way you treat your business data. Many installations have a strategy that includes daily, weekly, and monthly backups. Others also include quarterly and annual backups, which usually line up with fiscal quarter and year ends. Typically, the more infrequent the regular backup is, the longer the retention period is for the data. For example, monthly backups may be retained for 6 months and year end backups may be retained for 7 years.

It is good practice to include collected performance data in these backups.

If your business operations include cyclic processing such as end-of-month, end-of-financial year, and so on, it is likely your backup strategy is built around this. The intent is usually to take a snapshot of application databases before or after these events, in case the information needs to be analyzed as at those points.

The same is true of collected performance data. For example, you should retain, on offline media, data collected during month or year end processing. This can be useful for comparison purposes if future cyclic processing exhibits different performance characteristics.

When saving performance data, if the collector uses Management Collection objects, these objects can be saved rather than the database files. This will help minimize offline media usage.

8.8 Collecting Collection Services data for an entire server

Collection Services has the ability to collect high level performance data for all logical partitions on a single physical server. It achieves this by communicating with the Hypervisor, requesting partition configuration and utilization at each interval, and then storing that data in the Management Collection object. Like any other Collection Services data, the data can then be placed in a performance database for subsequent analysis. The data is stored in a file called QAPMLPARH.

Note: The Hypervisor collects the information about each partition regardless of which operating system or operating system level is running in the partition. For example, it will collect data for IBM i, Linux, and AIX.

The data can also be investigated using the perspectives under the content packages in Investigate Data in IBM Systems Director Navigator for i. Some of the available perspectives are:

- Logical Partitions Overview
- Donated Processor Time by Logical Partition
- Uncapped Processor Time Used by Logical Partition
- Physical Shared Processor Pool Utilization
- Physical Processors Utilization by Physical Processor
- Dedicated Processors Utilization by Logical Partition
- Physical Processors Utilization by Processor Status Overview
- Physical Processors Utilization by Processor Status Detail

An example of a Logical Partitions Overview perspective is shown in Figure 8-3.

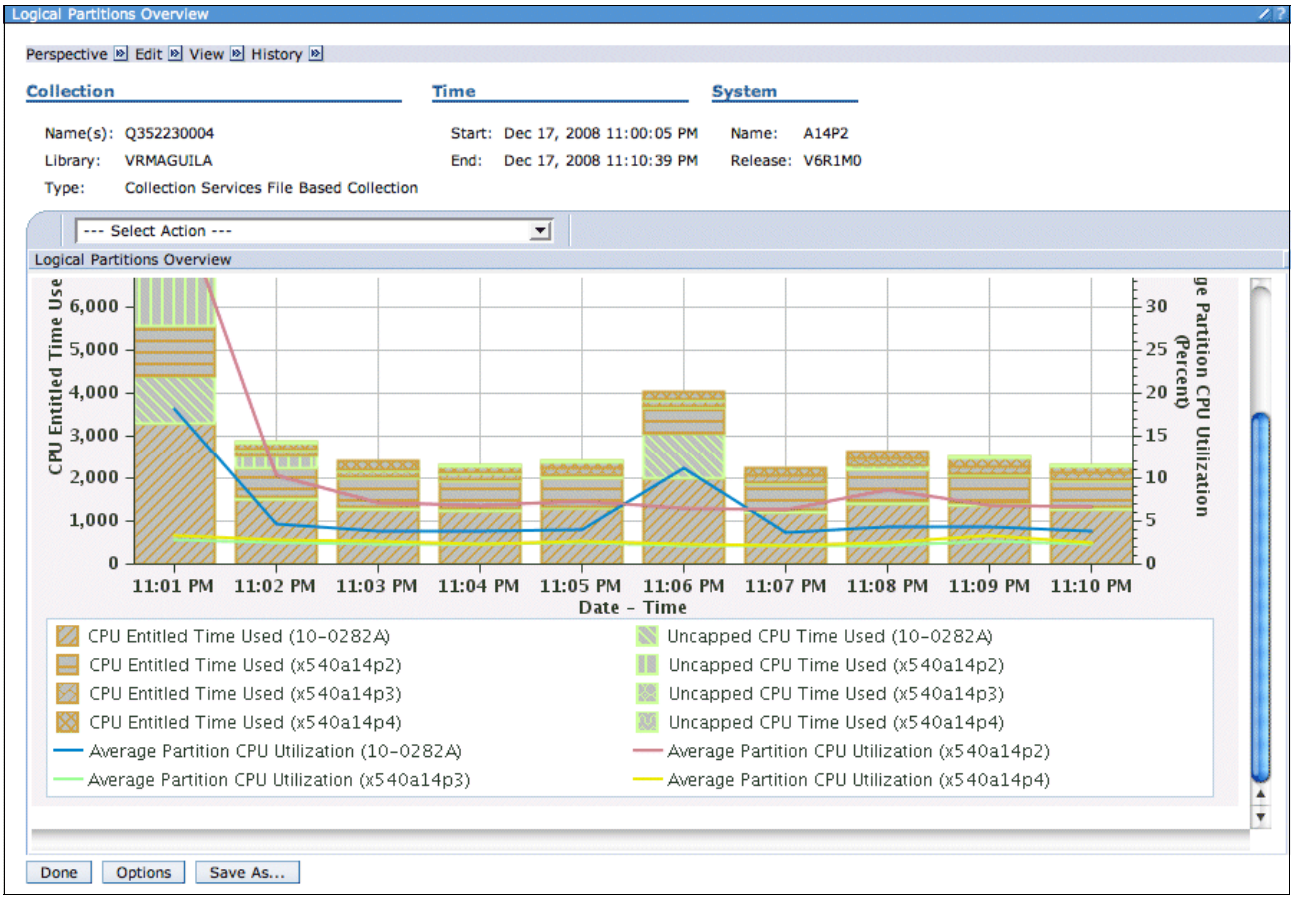


Figure 8-3 Logical Partitions Overview

The collection and investigation of data from multiple partitions is a powerful feature. Not only can you view total server utilization from an IBM perspective, you can include all other operating systems as well. None of the other operating systems currently have a similar ability to collect and investigate this data.

> **Note:** The feature is only available on POWER6 and above servers, with a minimum firmware level of xx340_061.

Activation of this feature is not done using IBM Systems Director Navigator for i. It must be done using a Hardware Management Console (HMC) or the Integrated Virtualization Manager (IVM). These two products are the standard interfaces to the Hypervisor. In a HMC, the feature is turned on or off by using the Allow performance information collection partition profile attribute, as shown in Figure 8-4.

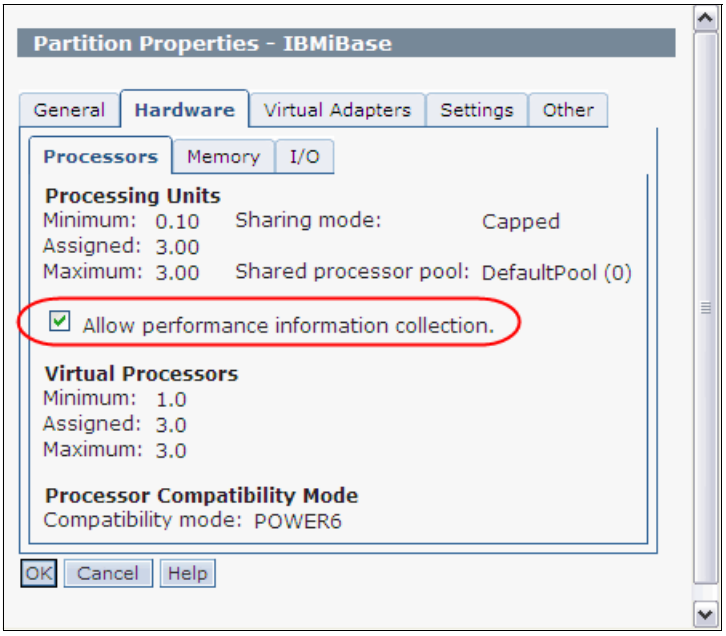


Figure 8-4 Allow performance information collection

For IVM, use the **`lssyscfg`** command, specifying the allow_perf_collection (Permission for the partition to retrieve shared processor pool utilization) parameter, Valid values for the parameter are:

0 Do not allow authority.

1 Allow authority.

The attribute only needs to be set in the profile for the IBM i partition that is intended to be used to collect the data from the Hypervisor. As well as passing the collected information to Collection Services in that partition, the Hypervisor will also source attributes, such as the interval duration, from the same partition. This does not alter the Collection Services attributes in other IBM i partitions; they continue to be set and used on a per partition basis.

8.9 Disaster recovery and other failover systems

An often overlooked but very important component of a business operation is the capacity of disaster recovery or failover systems. A common practice is to size a disaster recovery system smaller than a production system, based on the assumption that some businesses can still operate at a reduced performance level (or a reduced application portfolio) during a failover. In some cases, businesses replace their production system with a new system to boost performance and then reuse the retired production system as a disaster recovery system.

Extreme care should be taken when sizing disaster recovery systems.

If, for example, an old production system becomes the disaster recovery system, it is unlikely that system could handle a production workload, given that it was not suitable for normal operations in the first place and was replaced.

In a situation where a disaster recovery system is sized based on running a reduced application portfolio, sometimes this may not be practical. In many cases where multiple applications are hosted on a single partition for a given business, the applications are interrelated in some way. This could be a dependency of one application on a part of another application's database. Alternatively, there could be application interfaces that can cause the application to fail if the interface is not available.

Best practice suggests that disaster recovery capacity should equal production capacity. This can be an undesirable expense for many businesses. IBM has offerings, such as Capacity Backup for Power Systems, to help minimize the cost.

If, for business reasons, an undersized disaster recovery system is in place (or proposed), consider using a tool such as Workload Estimator to assess the performance of a collected production workload on that sized system. However, remember that the Workload Estimator is just that: an estimator. The only way to be 100% certain of the performance of a disaster recovery system is to regularly fail over to it and run a production workload for a period of time.

8.10 Web application performance

As mentioned in "Web application performance" on page 57, Collection Services can collect performance metrics for WebSphere Application Server jobs. Much of the data comes from WebSphere Performance Monitoring Infrastructure (PMI) data and transaction counters.

WebSphere Application Server also provides two tools that can be used to manage performance:

- Web Performance Advisor
- Web Performance Monitor

The Web Performance Advisor collects static data, for example, WebSphere container settings, and provides advice about the value of the setting. The 8.10.2, "Web Performance Monitor" on page 237 differs in that it collects real-time transaction data but does not provide any advice.

These tools are discussed in the following sections.

8.10.1 Web Performance Advisor

The Web Performance Advisor provides a way to view, evaluate, and modify the attributes that affect the performance of your Web environment. Clear definitions of the attributes are provided along with recommended values. The tool also provides a rating for each attribute to help guide the user to acceptable settings.

A Web environment is a grouping of related Web and application servers that form a Web solution. A Web environment is typically made up of a single WebSphere Application Server instance or profile and all the application servers contained within, its corresponding IBM HTTP Server for i instance, and any system attributes that could have a direct effect on the performance of the Web environment.

The Advisor function allows you to manage system attributes and to manage Web environment attributes. From the manage system and manage Web environment windows, you can view, evaluate, and change each performance attribute. While evaluating each performance attribute, click the attribute's **Advise** link to learn about the attribute and find the recommended setting.

An export function allows you to save existing performance settings in a performance profile. This profile can be evaluated, compared, or sent to a performance expert for analysis and modification.

Note: Web Performance Advisor is a complementary tool to Workload Estimator. Web Performance Advisor recommends values to tune the performance of your Web applications and environments. These recommendations are given without taking into account any specific runtime workloads on the system. Workload Estimator helps predict a system and the necessary physical resources for a given workload. Both tools may give different recommendations and may differ from the documented minimums. In addition, both tools attempt to ultimately provide the necessary information to best tune your Web environment and system performance. However, no representation is made by either tool that their corresponding recommendations will achieve a specific throughput, response time, or other performance measurement in any given installation environment. To use Workload Estimator to size your workload, see 7.2.1, “IBM Systems Workload Estimator (WLE)” on page 187.

Figure 8-5 shows an example of CPU savings achieved after amending settings as advised by the Web Performance Advisor. In the example, two different applications are shown, WebFacing and Trade[3].

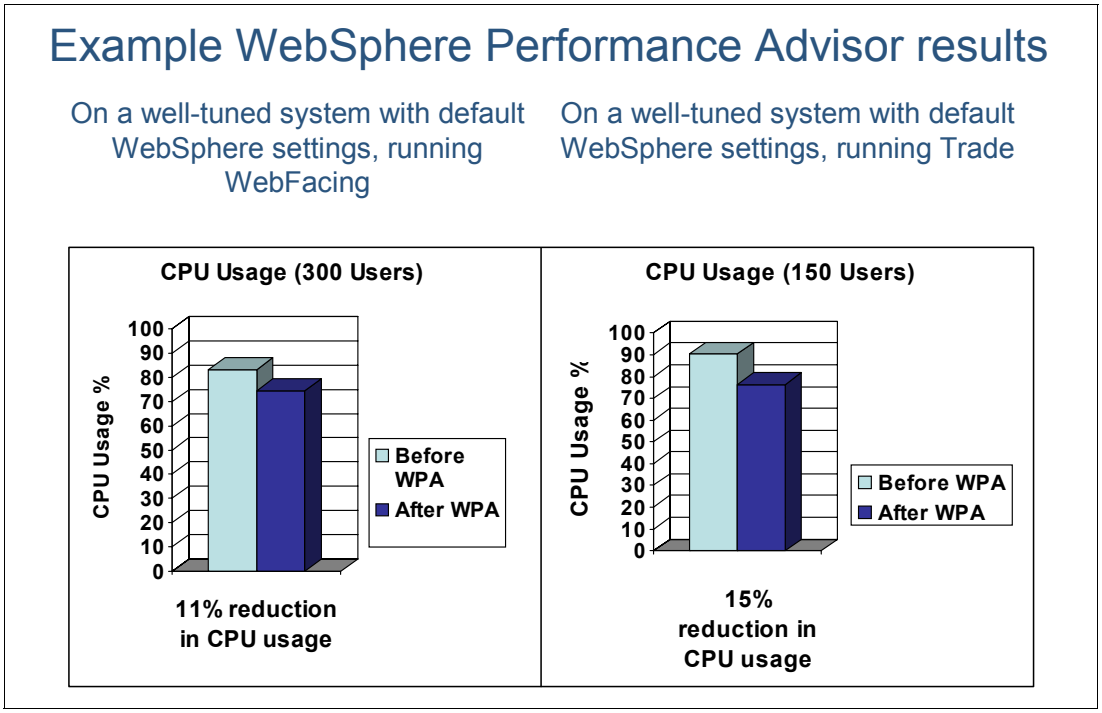


Figure 8-5 Example WebSphere Performance Advisor results

The detailed steps of using the Web Performance Advisor can be found by going to the IBM i V6.1 Information Center and selecting **Networking** → **HTTP Server** → **Tasks** → **Web tasks**.

8.10.2 Web Performance Monitor

The IBM Web Administration for i interface provides the Web Performance Monitor to monitor and evaluate the performance characteristics of your Web environment.

A Web environment is a grouping of related HTTP and application servers that form a Web solution. In a Web environment, you may have to monitor the performance of application servers, HTTP servers, databases, and applications jobs. The Web Performance Monitor provides a single interface to:

- Monitor your entire Web environment.
- Discover how your system time and resources are being spent.
- Identify potential bottlenecks.

Note: Only one Web environment on a partition can be monitored at one time.

[3] The Apache DayTrader Performance Benchmark Sample Application simulates a simple stock trading application that allows users to log in/log out, view their portfolio, lookup stock quotes, buy and sell stock share, and manage account information. Originally based on the Trade Performance Benchmark Sample Application for WebSphere Application Server, the application was donated to the Apache Geronimo open source community in conjunction with IBM's delivery of WebSphere Application Server Community Edition in 2006.

When the Web Performance Monitor is enabled, the **Transactions** tab displays the transaction statistics for jobs that are collecting Application Response Measurement (ARM) data in your environment. For each transaction, the average response time and the number of completed transactions information is shown.

ARM is an Open Group standard. For more information about the standard, refer to The Open Group Application Response Measurement (ARM) Web site, found at the following address:

http://www.opengroup.org/tech/management/arm/

Figure 8-6 shows how ARM is used to capture performance statistics at different points in a Web environment.

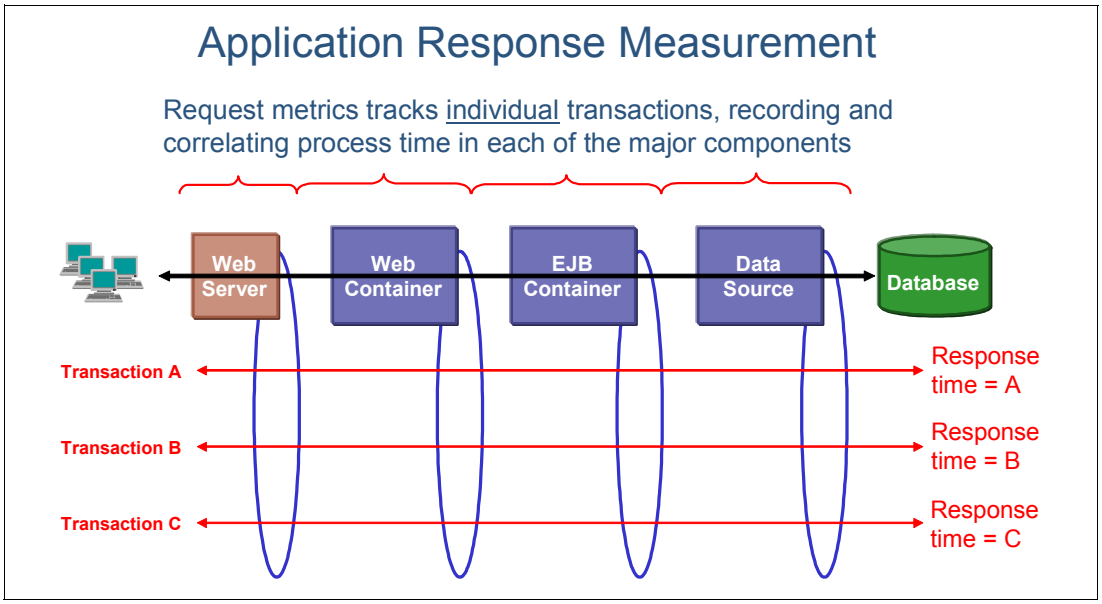


Figure 8-6 Application Response Measurement

In the Web Performance Monitor Transactions tab, you may also look at the transaction times for a single IP Address or host name that allows you ways to filter what is happening within an application. The transaction information shows which jobs are processing the requests for each user or IP address. This very specific user information can help identify which jobs may be slowing things down.

When the Web Performance Monitor is enabled, the Jobs tab displays the active servers and jobs in your environment. The Web Performance Monitor tool displays only the jobs on the system that are related to the selected Web server. For example, if the HTTP server is on a remote system, then there will not be any jobs for the HTTP server displayed. For each job, the current user, CPU usage percentages, run priority, number of threads, average transaction response time, and number of complete transactions is provided. Large results for average response time or CPU usage percentage may indicate a bottleneck and provide a starting point to investigate a performance problem.

The Web Performance Monitor is a part of the IBM Systems Director Navigator for i. The detailed steps of using the Web Performance Monitor can be found by going to the IBM i V6.1 Information Center and selecting **Networking** → **HTTP Server** → **Tasks** → **Web tasks**.

8.11 Performance Capabilities Reference

Performance improvements can often be gained by upgrading to more recent hardware technology or to new versions, or releases, of the operating system.

The *IBM Power Systems Performance Capabilities Reference IBM i Operating System Version 6.1* publication is intended to help provide guidance in terms of IBM i operating system performance, capacity planning information, and tips to obtain optimal performance about the IBM i operating system. The document is typically updated with each new release or more often if needed. The April 2009 edition is an update to the January/April/October 2008 edition, and reflects new product functions announced on 28 April, 2009.

The April 2009 edition includes new performance information about previously announced IBM Power Systems, including Power 520 and Power 550 utilizing 4.7 and 5.0 GHz POWER6 processor technology, respectively, and the IBM BladeCenter JS23 models, running IBM i and driven by POWER6 processor technology. The edition also includes new information about:

- The POWER6 model 570
- DB2 for i SQL Query Engine
- WebSphere Application Server, including WebSphere Application Server V6.1 both with the Classic JVM and the IBM Technology for Java (32-bit) JVM
- WebSphere Host Access Transformation Services (HATS), including the IBM WebFacing Deployment Tool with HATS Technology (WDHT)
- PHP - Zend Core for i
- Java, including Classic JVM (64-bit), IBM Technology for Java (32-bit), IBM Technology for Java (64-bit), and bytecode verification
- Cryptography
- Domino 7 and Workplace Collaboration Services (WCS)
- RAID6 versus RAID5 disk comparisons
- Added internal storage adapters
- Virtual Tape
- IPL Performance

The wide variety of applications available makes it extremely difficult to describe a *typical* workload. The data in the publication is the result of measuring or modeling certain application programs in very specific and unique configurations, and should not be used to predict specific performance for other applications. The performance of other applications can be predicted using a system sizing tool such as IBM Systems Workload Estimator.

The *IBM Power Systems Performance Capabilities Reference IBM i Operating System Version 6.1* publication can be found on the Performance Management on IBM i Web page at:

`http://www.ibm.com/systems/i/advantages/perfmgmt/resource.html`

A

Understanding disk performance metrics

This appendix contains the full text of a three part article written by Alexei Pytel of IBM, Rochester. It discusses:

- Why disk performance is important
- Introduction to disk operation on IBM i
- Metrics of disk performance

It has been reproduced here in entirety with some minor updates to formatting, product names, and so on.

Note: In the article, the author makes references to new functions that were introduced in IBM i V5.4. He also provides examples based on IBM i V5.4. IBM i V6.1 has introduced changes to the product documentation and function, so some content in this article is release dependent. The article has *not* been completely updated for IBM V6.1 in this IBM Redbooks publication.

© Copyright IBM Corp. 2009. All rights reserved.

Part 1: Why disk performance is important

The title of this part should really be *Why disk performance is important and how can I tell if I care*. In it, we discuss why disk performance can be very important in some environments and also how to determine whether one of those environments might be yours.

Why disk performance

Disk performance is one aspect of application performance (or the performance of a computer system in general), which often does not receive as much attention as it deserves. When thinking about performance, one usually thinks about processor performance first. Comparing gigahertz numbers is a staple of performance comparisons between different systems and platforms.

However, a processor can only work on the data that is located in main storage. If data is not there, the processor has to wait until data is brought there. And it is not doing anything particularly useful during this time. All processors, the fastest and the slowest, wait with the same speed.

For many years, there has been tremendous progress in processor performance. Modern processors operate at gigahertz speeds. On the other hand, there has been no comparable progress in the disk technology. Remarkable results were achieved, especially in the storage density and capacity of disk devices. Modern disk drives are a lot more compact and contain a lot more data than previously was thought possible. However, the characteristic time scale of events in the life of the processor is measured in nanoseconds. For comparison, the typical response time of disk accesses is still measured in milliseconds. This is a million times difference! A millisecond is an eternity compared to processor speeds. This explains why disk performance deserves more attention that it first appears.

Of course, not all applications are equal. Some applications are using mostly processor resources. For example, a program that performs complex calculations for a SETI project and only occasionally accesses a disk to save the intermediate results would probably not care much about disk performance.

On the other hand, an end-of-day process in the bank, which processes a database of several terabytes in size, may spend 95% or more of its run time waiting for disk accesses to complete. In this case, it clearly makes a lot of sense to spend some time looking at disk performance.

The obvious question is this: It is possible to easily tell if disk performance is important for my own application? IBM i may have an answer to that question.

Wait state accounting

To discover how much time your applications spend waiting for disk I/O operations to complete, we turn to information provided by IBM i wait state accounting.

Wait state accounting is a relatively new technology, so let us discuss it.

The IBM i operating system records, for every active thread, the time this thread spends in different states, that is, so much time spent using CPU, so much time spent waiting on database record lock, so much time spent waiting for disk I/O to complete, and so on.

There are many reasons for a thread to be waiting for something, so it is not practical to report every single reason separately.

To make wait state accounting more useful, wait reasons are grouped in a relatively small number of wait categories, or wait buckets. In IBM i V5.4, there are 16 such buckets.

Wait bucket information for each active thread is captured by Collection Services and is exported in two performance database files, QAPMJOBWT and QAPMJOBWTD.

Note: Collection Services is a component of the IBM i operating system, which is responsible for collecting all kinds of performance data. Collected performance data is exported in a set of database files, which are used by many other tools and components. Collection Services and its database files are documented in the IBM i Information Center; go to the Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Collection Services** and **Systems management** → **Performance** → **Reference information for Performance** → **Collection Services data files**.

The QAPMJOBWTD file contains information about the supported wait buckets and about the number assigned to each bucket. The QAPMJOBWT file contains information about the time spent by each thread in a specific wait bucket and how many times the thread was waiting for the reason associated with that wait bucket. Each bucket is identified by the number shown in the QAPMJOBWTD file.

Analyzing disk I/O time for a job

The following SQL query over QAPMJOBWTD file will show which wait buckets are available:

```
select * from qmpgdata/qapmjobwtd
```

Note: Collection Services files are multi-member files and SQL does not support multi-member files directly. The following OVRDBF command issued just before running the previous query will point SQL to the library and member where your performance data was collected:

```
ovrdbf qapmjobwtd qmpgdata/qapmjobwtd *first
```

Note: Bucket descriptions in the QAPMJOBWTD file are stored in Unicode. If your job was configured to run with CCSID 65535, you will most likely see gibberish. Before running this query, change the CCSID of your job to the CCSID associated with a primary language on your system, for example, 37 for US English.

Table A-1 provides a list of wait buckets supported on IBM i V5.4.

Table A-1 Wait bucket descriptions

Wait bucket number	Wait bucket description
1	Time dispatched on a CPU
2	CPU queuing
3	Reserved
4	Other waits
5	DASD (page faults)
6	DASD (other)
7	Socket
8	Idle / waiting for work
9	Seize
10	Object lock
11	Record lock
12	Gate
13	Java
14	Journal
15	Mutex and Semaphore
16	Reserved

To see the impact of disk I/O on performance, we look at the buckets DASD (page faults) and DASD (other). The former shows the amount of time that threads spend waiting for disk I/O operations caused by page faults, and the second one shows all of the other disk I/O operations. Note the numbers of these buckets (5 and 6).

This means that the time spent waiting for these events will appear in the fields JWTM05 and JWTM06 in the QAPMJOBWT file in the record, which contains buckets 1 to 16 (this record will have field JWBGN = 1).

The QAPMJOBWT file will have a record for each thread for each interval where this thread had any activity.

Let us take QSQSRVR jobs as an example and see how much time these jobs were waiting. Example A-1 shows how much time QSQSRVR jobs spent waiting for disk I/O operations to complete in relation to the total time of the job's activity.

Example A-1 SQL statement to summarize disk wait time (IBM i V5.4)

```
select jwname, jwuser, jwnbr,
          dec( sum(jwtm05+jwtm06), 8 ) as "Disk waits",
          dec( sum(intsec) * 1000, 8) as "Job act time",
          dec( sum(jwtm05+jwtm06)/( sum(intsec) * 10 ), 5,2) as "Dsk wt %"
from qmpgdata/qapmjobwt
where jwbgn = 1 and jwname like 'QSQSRVR%'
group by jwname, jwuser, jwnbr
```

Figure A-1 shows the result of this query.

```
                                        "Disk wait"   "Job act time"  "Dsk wt %"
                  Job         Job
Name              user        number
QSQSRVR           QUSER       010527   42,198,407     57,212,000       73.75
QSQSRVR           QUSER       010622          992        107,000         .92
```

Figure A-1 Summarized disk wait time

Where:

- The “Disk wait” column shows the total time that the SQL server job has spent waiting for I/O to complete (in milliseconds).
- The “Job act time” column shows the total time that the QSQSRVR job was active during the entire collection (also in milliseconds).
- The “Disk wt %” column shows the percentage of job time spent waiting for disk I/O.

As we see, for the second job, disk I/O is clearly not a problem; less than one percent of its lifetime was spent waiting for disk I/O. But the first job has spent almost three quarters of its time in disk waits. Therefore, even modest improvement in disk response time will have an immediate positive impact for this job.

Transactional jobs considerations

In our previous example, QSQSRVR jobs were doing batch type of work. This is why the previous query worked well, because batch jobs do not have to wait for input from an outside source, for example, from a human user. But what can we do about transaction-oriented jobs? It does not make sense to relate disk wait time to the entire lifetime of a job, because a job may be spending most of its time waiting for work.

In this case, it helps to relate disk wait time to some measure of work performed by the job. For example, for traditional interactive jobs, a good measure of disk performance will be the average disk wait time per transaction. The query shown in Example A-2 performs this action.

Example A-2 Selecting average disk wait time per transaction (IBM i V5.4)

```
select jwname, jwuser, jwnbr,
           dec(sum(jwtm05+jwtm06),8) as "Disk wait",
           dec(sum(jbntr),8) as "Trans",
           dec(sum(jwtm05+jwtm06)/sum(jbntr),4) as "Wait/trans"
from qmpgdata/qapmjobwt w, qmpgdata/qapmjobos o
where jwbgn = 1 and jwname like 'QPADEV%'
           and jwtde=jbtde and w.intnum = o.intnum
group by jwname, jwuser, jwnbr
```

Figure A-2 shows an example of the output of this query.

```
                                        "Disk wait"     "Trans"   "Wait/trans"
                  Job         Job
Name              user        number
QPADEV0005        JOE         012066        19825            50        396
```

Figure A-2 Average disk wait time per transaction

Where:

- The “Disk wait” column shows the total time that the interactive job has spent waiting for I/O to complete (in milliseconds).
- The “Trans” column shows the total number of display transactions performed by that job.
- The “Wait/trans” column shows the average time that the job was waiting for disk I/O operations per transaction (in milliseconds).

In the example shown, the job was spending, on average, about half a second (396 milliseconds) for each transaction waiting for disk I/O operations to complete.

Where to go from here

We can now discover how much time an application spent waiting for disk operations to complete. Looking at this data, can we decide if the numbers are good or bad? This is always a somewhat subjective decision, but there are several factors to consider.

For example, we have a large batch job that takes 10 hours to run and we know that 75% of that time was spent on disk waits. If disk response time can be improved by just 20%, this will reduce the job’s run time by an hour and a half. This is definitely something to think about when talking about a batch window.

Another example: If disk wait time contributes half a second to every transaction, any improvements in disk performance will be directly reflected in improved transaction response time.

So, you have reviewed performance data for your applications and decided that you have a disk performance problem on your system. What can you do next? The obvious next question to ask is, why is the disk slow and what can I do about it?

In the next section, I discuss the disk performance data available on IBM i and show some useful conclusions that can be made based on analysis of this data.

Part 2: Introduction to disk operation on IBM i

“Part 1: Why disk performance is important” on page 242 tried to demonstrate that disk performance is often a crucial aspect of total application performance. It also explained how you can query IBM i performance data files to decide if disk performance is really important for your own application.

Now suppose you ran the queries and decided, yes, my application does spend too much time waiting on disk operations. What shall I do next?

The next logical step is to try to understand why disk operations are taking as much time as they do and is there anything that can be done to improve them. To investigate this situation, we look at disk performance data reported by IBM i.

Before turning to the disk performance data, let us look at what actually happens when disk operation is performed on IBM i, which will help us better understand the meaning of those disk performance counters.

Basic flow

Figure A-3 a basic disk subsystem flow.

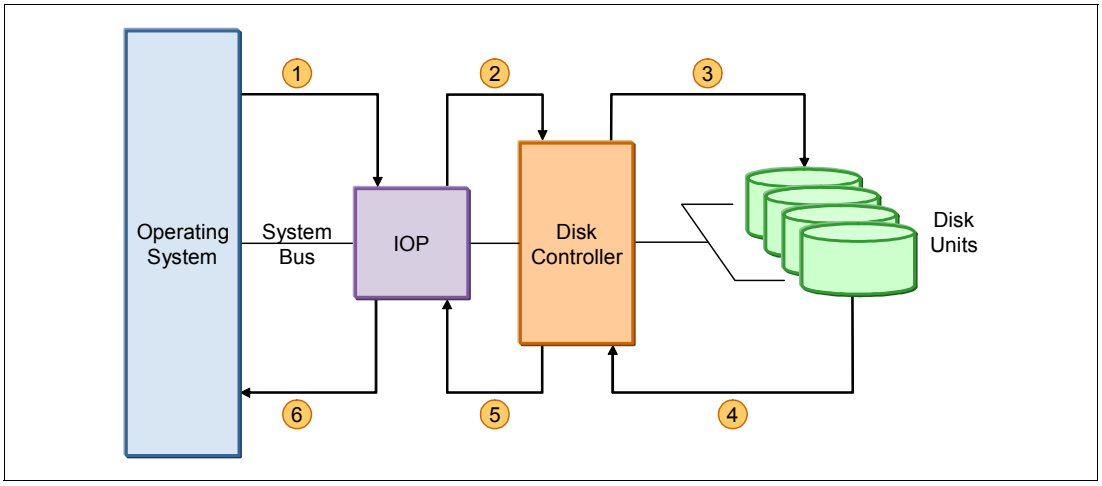


Figure A-3 Disk subsystem: basic flow

The operating system sends a disk operation across the system I/O bus to the input/output processor (IOP) (1). The IOP does its job and sends the operation down to the disk controller, also known as the disk storage adapter or the disk input output adapter (IOA) (2). The storage adapter manages disk units, which are attached to it by means of local buses. Having received the operation from the IOP, the storage adapter transfers it to the actual disk unit (3). The disk unit performs the requested operation and notifies the storage adapter when the operation is complete (4). The storage adapter notifies the IOP (5) and the IOP makes this situation known to the operating system (6). The disk operation is complete.

The storage adapter and IOP together manage the transfer of data between system main storage, where operating system code and user applications can access it, and a disk unit.

If another I/O request arrives for the same disk unit while the previous disk operation is still in progress, the new operation is queued by the IOP for later execution. If more disk operations arrive at the IOP than it has resources to manage, the disk operation queue will spill over to the operating system.

This picture describes the basic scenario and this is how disk operations were performed on the early systems.

Over the years, there have been many enhancements to this design, which had a profound effect on how disk operations are processed and on the performance of this process.

Storage adapter caches

The single most important change to disk I/O architecture was adding cache storage to the storage adapter. The purpose of the cache is to keep some data in the local memory of the storage adapter so that the disk operation would not have to access the actual disk unit every time.

A disk device essentially remains a mechanical device with moving parts. With all the advances in disk technology, the mechanical nature of a disk device still remains a defining factor in its performance.

All the components (bus, IOP, and storage adapter hardware) between the system and actual physical disk device operate at electronic speeds; this difference is very large indeed. With the ensuing processing and moving of data, the typical time to access data stored in a storage adapter cache is roughly about 0.1 milliseconds. At the same time, with current technology, a typical access time to the disk surface (which includes head positioning and rotational delays) is about 4 - 5 milliseconds.

This is a big difference. The purpose of the cache is bridge this gap.

Actually, there are two different caches in a storage adapter: a read cache and a write cache. They work closely together, but their impact on disk performance is very different.

Read cache

The basic idea of a read cache is that whenever data is read from a disk unit, it is saved in a storage adapter read cache so that it can be accessed quickly when it is accessed again, which leads to a gain in performance.

There are many strategies and algorithms to manage data in the read cache, which may have advantages and disadvantages for different patterns of disk access. But they all have to overcome two fundamental problems inherent in a concept of a read cache.

First of all, before a read cache starts improving performance, *the data needs to be read*. This may sound trivial, but before any piece of data can be found in cache, it has to be read from disk in the first place. The second and future accesses to this piece of data will be fast cache operations, but that very first operation still has to be a slow read from a disk surface. There are sophisticated cache management algorithms that try to solve this by making a guess at which data will be read next and reading it before any application actually requests that piece of data. But this logic depends on the application's pattern of disk access. If the wrong guess is made, the cache space and precious device throughput are wasted on unneeded data.

Another problem is that in order for any performance gain to be achieved, *the same data needs to be requested at least twice*. For example, imagine an end-of-day batch process in a bank or an insurance company which processes a huge database and applies some kind of financial transactions to many accounts. The database could be terabytes in size, and for hours it is processed in one direction; no records (or very few records) are accessed more than once. A read cache is there, but it does not help much. In fact, the very task of managing the cache introduces a certain impact of its own, so for those kind of workloads the disk performance with a read cache could be even somewhat worse than without it.

Thus, the performance impact of a read cache is variable. In particular, increasing the size of a read cache may improve performance or may do nothing or may even degrade the performance. It all depends on the behavior of the workload, the pattern of data access, and the size of the database accessed by applications. The same disk configuration can demonstrate a stellar performance with one application but will not work so well for a different one.

Specifically, on IBM i, where entire system main storage acts as a cache for system objects, the influence of a read cache tends to be less dramatic than it might be expected looking at other systems' experiences.

Write cache

The write cache works entirely differently than the read cache. In fact, it would be more correct to call it a *write buffer* rather than write cache, except that it works like a cache when read operations access it for data.

When a disk write operation is requested, the data is written to a memory buffer and is queued to be written out to a disk device at a later time (this later operation is called *destaging*). After a buffer is queued, disk operation completion is signaled. As far as the waiting application is concerned, the write operation is over.

Of course, data is not on a disk surface yet, so what happens if a system loses power before destaging is complete? In this situation, a write cache is protected by an independent source of power, a battery, which keeps write cache contents intact until power is restored and data can be evacuated to its proper place on a disk surface. This is called non-volatile storage (NVS).

Unlike a read cache, a write cache does not have any initial cache loading penalty and it does not depend on the workload behavior to be effective. A write cache helps bridge the performance gap between the mechanical speed of disk devices and electronic speed of other system components. Write cache has a particular benefit for RAID 5 and RAID 6 configurations. It takes four disk accesses to perform a write to a RAID 5 protected disk device. It takes six disk accesses for RAID 6. But with a write cache, an application does not care about all these disk accesses. This makes a write cache a must have feature when RAID protection is used.

However, all those writes to a physical disk device (destaging operations) still have to be performed. A write cache absorbs the disk write activity until physical disk device is overwhelmed to the point where the destaging operation queue backs up and there is no more memory left in a buffer pool to accept new write operations from system. Now new disk operations have to wait until some earlier operations are complete and free some memory. This is called a *write cache overrun*.

Although read operations do not queue up behind destaging operations, they still have to wait for the overloaded disk device to complete the current operation in progress. This will cause disk read operations to take longer as well.

Applications, which up to this point enjoyed sub-millisecond response time, suddenly experience response times that are much, much longer. This can be very dramatic. Without cache, disk response time steadily and noticeably degrades as disk operation rate increases. We have plenty of warning before this situation becomes really bad. With cache, disk response time degrades at a much slower rate, but when physical disk devices behind the cache reach a saturation point, there is a sudden dramatic increase in operation response time.

Impact of different disk protection levels

The impact of different disk protection levels is especially important to consider when choosing a disk protection level.

For example, we have an unprotected disk configuration with a certain number of disk units in it. There is a set of applications that produce a certain rate of disk operations, which translates to a certain number of disk operations per physical disk device. Let us see what different disk protection levels will do for that configuration in terms of disk performance.

If we start mirrored protection, this will halve the number of logical disk units. With the same application rate, this means that the write operation rate per physical disk device will double. However, the read operation rate per disk device will not change, because now there are two copies of the data and that data can be read from either of them.

If we start RAID 5 protection, the number of disk units will remain the same. This means that the disk operation rate per disk unit will not change. However, while each read operation issued by the system will translate into one disk access, every write operation will now translate to four disk accesses. This will create a much larger load on each physical disk device.

If we start RAID 6 protection, this will not impact the operation rate per device, as with RAID 5. However, every write operation will now translate to six disk accesses. This means that each physical disk device will have to work that much harder compared to RAID 5.

This topic is discussed in more detail in the white paper “Converting iSeries Disk Arm Requirements Based on Protection Type”, which is available at the following address:

http://www.ibm.com/systems/resources/systems_i_advantages_perfmgmt_pdf_diskarmc.pdf

Parallel disk operations

Another important change to a basic flow was parallel operation support by the disk storage adapters. Today’s disk storage adapter does not process disk operations one by one. Instead, storage adapter microcode is able to process a disk operation even if another one is already in progress.

Together with a cache, this really helps to reduce the disk operation response time. Before that, even if disk operation could be entirely satisfied from cache, it had to wait before the previous operation was complete, which could be a slow operation requiring access to a disk. Because of this, disk response time tended to be dominated by the slow response of a disk device.

Now, as soon as a new operation arrives, the storage adapter microcode will determine if it can be satisfied from cache. If it can, it is serviced immediately and the result is sent back to the system. Only if an operation requires actual access to a physical disk device will it be queued behind the current operation in progress.

If there are too many fast disk operations, they will also queue up. It is important, however, that fast cache operations will only queue up behind other fast operations, and this is a much shorter wait.

External disks versus internal disks

External disk subsystems are attached to servers by way of Fibre Channel adapters. Figure A-4 shows how external disk attachment changes the basic picture.

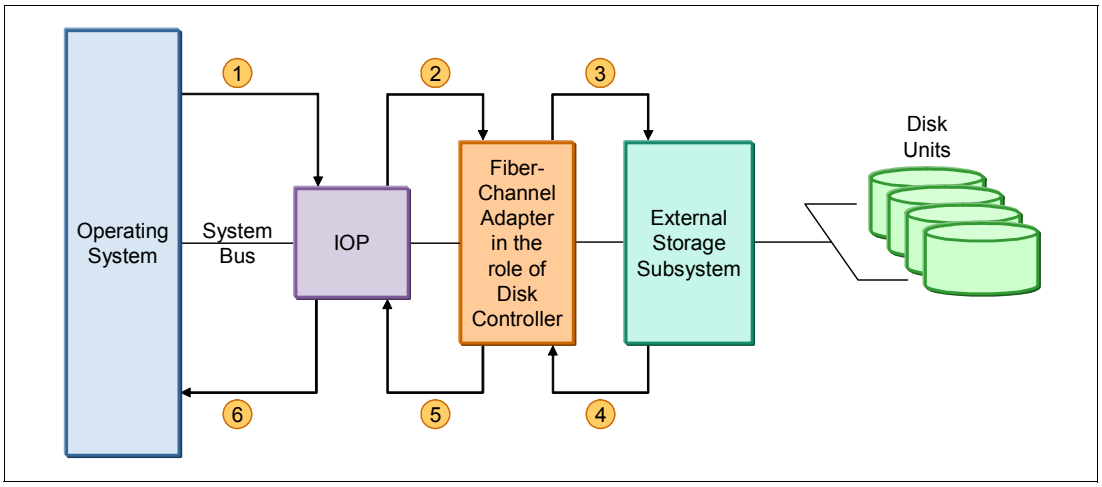


Figure A-4 Disk subsystem: external disk

Here the Fibre Channel adapter (host bus adapter or HBA) takes the place of a storage adapter card. Architecturally, it still plays the role of a disk controller.

The external storage subsystem at the other end of the fiber cable provides a set of disk units to the operating system. The fact that these disk units are not real, that is, they are logical units (LUNs) created by software running in an external storage control box, does not change how IBM i perceives those disk units.

As far as IBM i is concerned, the picture is very much the same as shown in Figure A-3 on page 247; there is still a disk controller (the Fibre Channel attachment card) and there are disk units managed by that disk controller.

The operation flow remains essentially the same, but there are some important differences. The most obvious difference is that the external storage subsystem is a computer system in its own right. It has its own processor resources, main storage, operating system, Fibre Channel adapters, disk adapters with disk devices attached to it, and so on. This computer system has its own configuration tasks, its own interfaces, and its own performance tuning challenges. Although all this has a very real impact on disk performance for IBM i applications, this topic is beyond the scope of this appendix.

The path the disk operations and disk data take between the actual physical disk devices attached to the external storage subsystem and IBM i is longer and more complicated than for internal disk units. An external subsystem can be attached directly by way of point-to-point fiber cable, but it can also be attached by way of a storage area network (SAN). This network has its own switches, multi-hop routes, and so on. This increases flexibility, but it inevitably adds some network latency. Additional processing will happen in the external storage controlling software.

Taken together, these delays add up and although this works at the electronic speeds, the raw performance of the external storage subsystem tends to be somewhat worse than the internal disk one.

The IBM i side system shown in Figure A-4 on page 251 has no knowledge about the internal workings of the external storage subsystem. In particular, the microcode running inside the Fibre Channel adapter cannot “peek” into the cache owned by that subsystem; it just sends each operation across the fiber cable to the external subsystem.

This is important because when a new request arrives from the system, the storage adapter cannot decide whether this is a fast operation that can be satisfied from cache or a slow operation that requires the physical disk drive (they all queue up together). Because of this situation, the disk response time tends to be dominated by the response times of slow “device” operations. This is especially important for workloads that request disk operations in bursts. The bursts tend to increase the average disk response times.

Multi-path support

One of the options for external storage attachment is multi-path support, where the same external storage subsystem can be accessed by way of different communication paths using more than one Fibre Channel attachment card. the operating system will recognize that disk units behind different disk controllers are actually the same logical disk unit.

Multiple paths are used for two purposes:

- They improve reliability. If one of the paths fail, the system is able to continue using the additional paths that remain available.
- Another benefit of using multiple paths is better performance. Additional paths increase the communications bandwidth between the system and external storage subsystem, which helps if network bandwidth is a bottleneck. In addition, multiple parallel paths reduce internal queuing. Multiple paths create multiple queues, which share the load. Reduced queuing improves the response time.

IOPless storage adapters

Originally, many years ago, electronic components were expensive and were concentrated in the IOPs. Disk controllers were more or less dumb devices, and were little more than electrical adapters between IOPs and the disk units they controlled. As electronic technology improved, storage adapters became more and more “intelligent”.

These “smarter” storage adapters allowed IOPs to become multifunction IOPs. A single IOP card was now able to control many controllers, not only disk controllers, but also communications controllers, tape controllers, and so on.

Eventually, storage adapter became capable of interfacing directly with the operating system without an IOP acting as an intermediary. The IOP has become a kind of a communications adapter between the operating system and disk storage adapter.

IOPs were eventually eliminated for the newest disk storage adapters. The remaining processing inside the IOPs was either moved to the disk storage adapters or the operating system itself. Performance-wise, due to the long evolution of IOP and storage adapter technology, this was not so much of a change than it might seem.

Indeed, recent disk performance benchmarks that compare disk performance of IOPless disk configurations to that of similarly configured IOP-managed ones did not show statistically significant difference. Refer to Chapter 14, “DASD Performance “, in *IBM i V6.1 Performance Capabilities Reference - April 2009*, which can be found on the Performance Management Web page at the following address:

http://www.ibm.com/systems/i/advantages/perfmgmt/resource.html

Figure A-5 shows how the original architecture has changed compared to Figure A-3 on page 247.

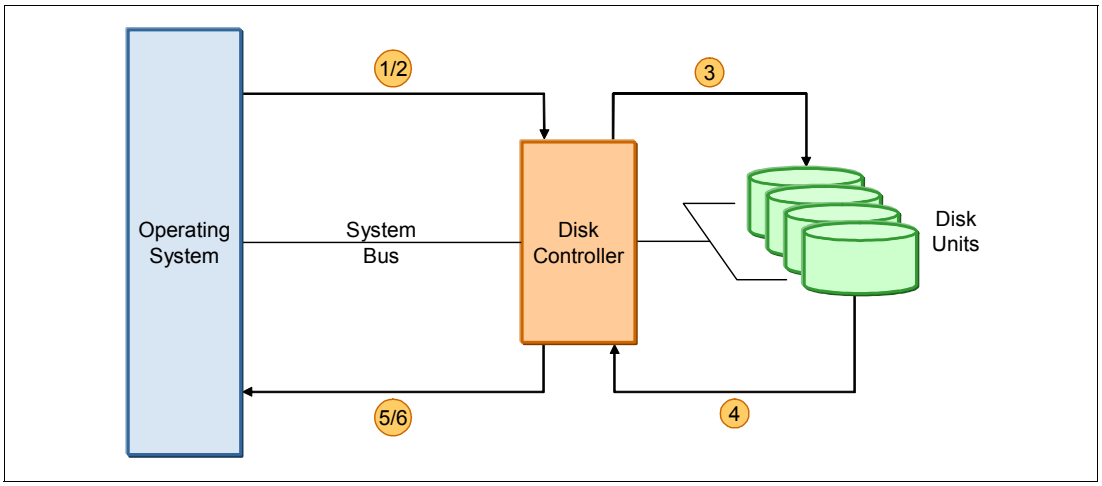


Figure A-5 Disk subsystem: IOPless disk

Basically, the flow between operating system and disk storage adapter (1) is a shortcut of what used to be a two step process, that is, (1) and (2) on Figure A-3 on page 247.

Hopefully, this part of this appendix has given you a better understanding of how disk operations are processed in different environments. This will be useful in “Part 3: Metrics of disk performance” when we look at the disk performance data reported by the operating system.

Part 3: Metrics of disk performance

In “Part 1: Why disk performance is important” on page 242, we explained how you can query IBM i performance data files to decide if access to data stored on disk has a significant impact on the performance of your application.

“Part 2: Introduction to disk operation on IBM i” on page 246 was dedicated to a discussion about how disk I/O operations are performed by IBM i in different disk attachment scenarios.

The purpose of this part is to examine IBM i performance metrics that provide a detailed view of disk performance. Analysis of these metrics may shed a light on possible performance problems and may suggest potential actions that will improve disk performance.

Disk performance data

A simple query of a job wait state accounting file (QAPMJOBWT; refer to “Analyzing disk I/O time for a job” on page 243) shows you how much time a job has spent waiting for disk I/O to complete. If this time comprises a significant part of a batch application’s run time or online transaction response time, you should probably rectify this situation.

However, job wait state accounting data does not provide any details as to what is the contribution of specific disk units, what could be the reason for poor disk response time, what are the components of disk response time, and so on. For this kind of data, we must look at another IBM i performance database file, QAPMDISK.

Note: Data found in the QAPMDISK performance database file is documented in IBM i Information Center. Go to the Information Center and select **Systems management** → **Performance** → **Reference information for Performance** → **Collection Services data files** → **Collection Services data files containing time interval data** → **Collection Services data files: QAPMDISK**.

The QAPMDISK file contains raw performance counters that are applicable to disk performance. This file has a record for each disk resource for every performance collection interval. A disk resource is one unique path between the operating system and disk unit. Internal disks have one single path to each unit. External disk units may have more than one path.

Performance data collected by the Collection Services (including the QAPMDISK file) represents interval summaries, so we can only talk about averages, not the execution details of individual disk operations. That would be the task for performance trace tools, which is beyond the scope of this appendix.

Tools are available that massage the raw data from the QAPMDISK file and show higher-level metrics to a user. For example, the IBM Performance Tools licensed program includes disk performance data in many reports it produces. However, for the purposes of this discussion, we concentrate on how higher-level disk performance metrics can be calculated directly from the raw data in QAPMDISK file.

The reader can compare this discussion with the reports produced by the Performance Tools licensed program and hopefully will find many similarities.

Note: The Performance Tool for i licensed program is documented in the IBM i Information Center. Go to the Information Center and select **Systems management** → **Performance** → **Applications for performance management** → **Performance Tools**.

Metrics of disk performance

In the following sections, various metrics of a disk performance will be described, including how they can be calculated based on the performance data from the QAPMDISK file. Most of the discussion in these sections is conceptual. To keep things simple, formulas used in this section will not take into account some finer details of data organization in the QAPMDISK file, introduced, in particular, by multi-path configurations. At the end of this appendix, you will find an SQL query that returns all the metrics discussed in this article and takes into account all of this complexity.

Note that the discussion in this appendix is not meant to be exhaustive.

Some general considerations that apply to most metrics:

- Metrics discussed in this article are all averages. There are many ways to look at average metrics: average for a specific interval for a single disk unit, average for all disk units in the interval, averages for all disk units in a disk pool over the time range of several hours, and so on.
- Statistical averages are an important tool, but remember that averages can be deceiving. Given sufficient amount of data, “good” results can easily mask “bad” results. For example, on a big system with several hundred disk units, a couple of disk units with bad response times will not noticeably change the overall average response time, which makes it easy to overlook a problem.

- IBM i stripes system objects across all available disk units. This is a good practice which in general improves disk performance (many SAN systems use this setup now). However, striping makes it hard to control the co-location of different system objects on disk. If there is a poor performing disk unit, there is a high probability that it will have an impact on many applications. Even a single poorly performing disk unit can slow down an entire system. But system-wide averages will not reveal that fact.
- Averages are fine when used carefully, but do not forget to keep an eye on individual disk units and individual time intervals, especially if you have reasons to suspect a problem.

Disk response time

Disk response time is the starting point for any disk performance analysis. It shows how long it takes to complete disk I/O operation, from start to finish.

Until IBM i V5.4, individual disk operations were not timed by system, and response time was calculated based on Little's Law:

Average Number of Requests on Server = Arrival Rate * Average Response Time

Note: *Server* is used here in the sense of a queuing theory, that is, this is a disk subsystem with IOP, storage adapter, and disk devices attached to them.

"Number of requests" includes disk operations in progress plus operations that are waiting in a queue. All the ingredients for this formula are available in the QAPMDISK file. Average disk response time in milliseconds can be calculated as:

(DSQUEL * INTSEC * 1000) / (DSSMPL * (DSWRTS + DSRDS))

You will notice that the number of requests is obtained by way of sampling (DSSMPL is the number of samples made during the interval0. To be accurate, the sampling technique requires a sufficient number of samples or a sufficient number of events sampled. The sampling rate is not particularly frequent (twice per second). The practical implication of this situation is that the response time calculated according to this formula is only valid if there were a significant number of operations issued to the disk unit. For a relatively idle unit, the formula can produce widely erratic results.

Note: This is one of the reasons why disk capacity planning should never be based on a run with an insignificant disk load.

In IBM i V5.4, the operating system directly times disk I/O operations. This new data is now added to the QAPMDISK file. When these new counters are available, the disk response time can be calculated much more accurately using this formula:

(DSSRVT + DSWT)/(DSWRTS + DSRDS)

Note: This new function was added by way of a set of PTFs in IBM i V5.4. The PTFs were included in recent cumulative packages, so if you are current with the PTFs, chances are that you already have this new function installed.

The new fields provide a much more accurate measure of disk response time, but there is a difference. The “old” data was measured by the IOP code. The “new” data is measured before operations are sent to IOP for processing, so it includes the delays between the IOP and the operating system. In general, this leads to more useful data, because the “new” response time is closer to actual delays experienced by applications.

However, there is a catch. The processing of disk operations on IOP is not impacted by what is going on inside the partition, or by the partitioning process itself. The “new” response time now includes delays caused by switching partitions on and off the physical processors, and also delays caused by CPU contention inside the partition. The latter ones are not an issue for most systems, except systems with extreme CPU contention. But the hypervisor delays can easily add a few milliseconds to a response time, especially for micro-partitions (partitions with very small processing capacity).

This consideration also applies to IOPless disk units. Such units do not have real IOP, so even if response time is calculated by the “old” formula, the result still represents a system view of disk operation delays.

What is the recommended guideline for “good” disk response time? There is no easy answer to that question. There are too many factors: disk technology, cache friendliness of a disk workload, mixture of reads and writes, tolerance of the particular application to the disk delays, and so on.

As a rule of thumb, with current disk technology, an average response time of less than 5 milliseconds is good, between 5 and 10 milliseconds is normal, above 10 milliseconds requires analysis, and above 100 milliseconds is bad.

Observation about averages applies to disk response time even more than to other performance metrics. Watch the disk response time for individual disk units at specific collection intervals. Bad response times that are significantly different from averages can be an indication of a storage adapter with some kind of a local problem (for example, a failed cache battery).

These response times can also be an indication of a hot spot, a disk location that, for some reason, gets more than its fair share of accesses. One reason for the hot spot could be the system configuration has an unbalanced distribution of data across disk units in the ASP. It makes sense to try disk balancing tools; they may do a good job of removing a hot spot. Refer to the man page for the STRASPBAL command for more information.

A hot spot can also be caused by application logic, which causes an unusual number of accesses to some common piece of data. This kind of hot spot is much harder to investigate and correct. There are tools to trace individual disk operations and determine which job issued them and which system objects were accessed. One such tool is Performance Explorer (PEX). Another is Disk Watcher.

PEX documentation can be found in the IBM i Information Center by going to the Information Center and selecting **Systems management** → **Performance** → **Applications for performance management** → **Performance explorer**.

Disk response time buckets

Good response times masking bad response times in good-looking averages is a pervasive problem, so a new function, disk response time buckets, was added recently to IBM i V5.4 by way of PTFs.

As the system measures response times for disk operations, they are classified into six ranges or buckets, which are operations that took less than 1 millisecond, 1 to 16 milliseconds, 16 to 64 milliseconds, 64 to 256 milliseconds, 256 to 1024 milliseconds, and those above 1024 milliseconds.

This data is reported in new fields in the QAPMDISK file. Each bucket is represented by three fields: number of operations which fall within the bucket, total response time of these operations, and total service time of these operations (look for fields DSBKCT*xx*, DSBKRT*xx*, and DSBKST*xx*).

Note: Response and service times reported in the buckets are measured at the system level and are consistent with the DSSRVT and DSWT fields.

Disk percent busy

This is probably the most often used measure of the health of a disk subsystem. This metric shows a percentage of the time that the disk was busy processing disk operations. This metric is also often, incorrectly, called *disk utilization*. Disk percent busy and disk utilization were the same thing when disk units were capable of processing only one operation at a time. Now the execution of two or more disk operations can overlap. Technically, if a disk is 100% busy, it may still be able to perform additional disk operations, in addition to those that are already in progress. We cannot really say that disk utilization is 100% in this case.

Note: It must be mentioned that there is another approach to disk utilization. Some operating systems report utilizations in excess of 100%, for example, a disk utilization of 500% means that there were an average of five operations in progress for this disk unit. IBM i does not use that approach.

However, this metric is a convenient way to tell at a glance if there is a potential disk performance problem. If a disk has too many operations sent to it by the system, or if disk operations are taking too long to complete, all of this will impact the disk and will be reflected in the percent busy metric.

The disk percent busy metric is calculated as:

100 * (DSSMPL – DSNBSY) / DSSMPL

You will notice that this data can also be obtained by way of sampling, with all implications already discussed. Disk units with light loads may demonstrate wide variations in business calculated by this formula.

Note: This is the same metric shown on the WRKDSKSTS screen.

Which value of this metric is considered normal? All performance guidelines are to a certain extent arbitrary. The rule used to be to keep disk percent busy below 40%. Benchmarks show that with the faster disk hardware and with ability of new storage adapters to process more than one operation in parallel, acceptable response times can be up to 70%. IBM Workload Estimator uses significantly more conservative guideline of 25%. This number is heavily biased towards OLTP environments, which are more sensitive to variations in disk response time. It also makes room for a natural workload growth. Batch-oriented workloads may be able to tolerate much higher disk utilizations.

Disk service time and disk wait time

Service time and wait time are components of the disk response time. When response time goes up, it is interesting to know whether the time is spent actually performing disk operations or while waiting for a queue.

The balance between service and wait times is closely related to the disk utilization: as utilization goes up, the wait component of the response time grows. Quite often, results of a queuing theory are applied to disk performance, for example, a queuing multiplier that relates service time and response time.

A word of caution: The simple and most commonly used model (known as M/M/1 in queuing theory) is based on certain assumptions about behavior of a “server” (disk unit). Years ago, these assumptions could be applied reasonably well to disk performance. However, this is no longer the case. It is theoretically possible to build a more realistic mathematical model for the behavior of the disk unit, but such model will be too complex for practical use.

The usual way to estimate average service time is by dividing the disk busy time by the number of operations executed during this time. This leads to the following formula (the result is in milliseconds):

((DSSMPL - DSNBSY) * INTSEC * 1000) / (DSSMPL * (DSWRTS + DSRDS))

The wait time will be, of course, the difference between disk response time and disk service time.

There are two problems with this formula. The first, previously mentioned, is that disk busy time is determined by relatively infrequent sampling, which yields unreliable results at low disk loads. The second problem is that this formula does not take into account the overlap between disk operations. Because of this, this formula will underestimate service time, and therefore overestimate wait time. One should keep that in mind when looking at the results shown by various performance tools.

To overcome these deficiencies, in IBM i V5.4, the operating system directly measures the components of each disk operation. Two new fields in the QAPMDISK file report total service time and total wait time of all disk operations.

So average service time becomes simply:

DSSRVT / (DSWRTS + DSRDS)

And average wait time becomes:

DSWT / (DSWRTS + DSRDS)

However, these new counters are measured at the system level, not on the IOP level. Service time measured at a system level will include operation queuing at the storage adapter level. Accordingly, wait time measured on a system level does not include waiting in a storage adapter queue. “New” formulas will therefore overestimate service time. The “real” service time experienced by storage subsystem is somewhere in between.

The wait time component of the disk response time will naturally grow as disk utilization grows, according to the results of a queuing theory. However, it can also be an indication of a burst workload. The burst of disk operations arriving very close together will cause excessive operation queuing even if disk utilization is low overall. To improve disk performance for burst workloads, you may try to distribute data over as many disk units as possible. This is easier to

achieve for external disk storage, where you can reconfigure the same physical devices to report to the system as a larger number of smaller logical units.

Read cache hit ratio

Read cache hit ratio shows, usually as a percentage, how many read operations were satisfied from cache. Going to the data saved in cache is much faster than going to the data on a disk surface, so obviously we want this number to be as high as possible.

There are two sets of cache statistics reported in the QAPMDISK file: controller cache counters and device cache counters. For internal disk units, both caches are reported. However, device cache, which is located on a physical disk drive itself, is relatively small and does not have much impact on performance. This is why in practical cases we usually consider only the controller cache.

For internal disk units, the read cache hit ratio is:

100 * DSCCRH / DSRDS

> **Note:** If data shows a reasonable number of read operations (DSRDS), but the cache hit counter (DSCCRH) is consistently zero, this is an easy way to tell at a glance if a storage adapter has any cache.

For external disk units, a storage adapter (architecturally a controller) is just a communications adapter that does not have any cache. The cache that resides in a storage subsystem is reported as a device cache.

Therefore, for the external disk units, the read cache hit ratio is:

100 * DSDCRH / DSDROP

Obviously, no cache at all or a very small cache is not good for performance. However, with a reasonable cache size and low read cache hit ratio, it is hard to predict if adding more cache will improve performance. This requires a complex analysis of the workload and data access pattern of the application. It might be easier to just add more cache and see if it improves performance.

In general, read cache hit ratio tells more about characteristics of a workload than about the functioning of disk units. It is not unusual to see a very small read cache hit ratio even with a very large cache if an application processes a great deal of data and rarely accesses the same data more than once. (See “Part 2: Introduction to disk operation on IBM i” on page 246 for detailed discussion of cache behavior.)

Write cache overrun ratio

The write cache overrun ratio shows, usually as a percentage, how many write operations had to wait for the destage operations to free up buffer space in the storage adapter memory. Writing data to the buffer space is many times faster than accessing a disk, so we want this number to be as small as possible (ideally zero).

For internal disk units, the write cache overrun ratio is:

100 - DSCCFW * 100 / DSWRTS

DSCCFW shows how many operations were *fast writes*, that is, those that did not have to wait for a buffer space.

> **Note:** If the DSCCFW counter is consistently zero, there are two possibilities. One is that this particular storage adapter has no cache. Another common situation is when the adapter does have a cache, but its write cache is inhibited, because the adapter cannot guarantee that its content is preserved in case of a system failure. Usually this happens when a cache battery is dead.

For the external disk units, the write cache overrun ratio is:

100 - DSDCFW * 100 / DSDWOP

The write cache is crucial for the performance of write-intensive applications, especially when database journaling is involved. In fact, write cache is a must for RAID configurations. It is hardly possible to have any decent disk performance with RAID without a write cache.

Total disk device operation rate

A cache can usually successfully shield an application from a slow mechanical device behind it and provide a much better response time than it would be capable of without a cache. However, eventually the disk still has to be accessed to read or write actual data. Every disk device has limits to its throughput. When this limit is approached, the deterioration of disk performance can be sudden and dramatic.

This is especially true in RAID 5 and even more so in RAID 6 configurations, when disk devices have to carry the additional burden of parity operations.

This is why it is important to keep an eye on the total disk operation rate, which can be calculated as:

(DSDWOP + DSDROP) / INTSEC

This metric is only applicable to internal disk units. External disk units are logical units that are mapped to physical disk devices by an additional layer of software inside an external storage server. To discover the actual load on the physical disk devices in an external storage server, you will have to use built-in tools provided with the storage server.

What is the acceptable disk operation rate for a physical disk? It depends on several factors, such as disk technology, of course, but also the pattern of access to the disk surface. In particular, long seek distances will make the disk unit work that much harder.

> **Note:** A hint: Seek statistics are also available in the QAPMDISK file. Look for the DSSK*n* field.

As a rule of thumb, with current disk technology, the maximum operation rate is in the range of 120-150 seeks per second. This is a maximum; you do not want to be anywhere near that for an extended period of time.

Note: In RAID 5 or RAID 6 configurations, a disk device has to perform parity operations for other disk devices in the same RAID set. This is usually not noticeable because these operations are balanced out as different disk units are accessed. However, if disk units from the same RAID set are allocated to different ASPs, you may be surprised to see that some disk units in the otherwise idle ASP report activity; this is work they have to do for disk units from another ASP.

Write cache efficiency

As data is sitting in the write cache waiting to be destaged to the disk, cache management logic tries to apply all kinds of optimizations to reduce the actual number of disk accesses. For example, a series of writes to the adjacent sectors can be coalesced into a single multi-sector operation. Multiple writes to the same sector may result in multiple updates to the data in cache, before data is ever written out.

The purpose of the write cache efficiency ratio metric is to show how successful the cache software was at this task. For example, if write cache efficiency is 15%, it means that the number of write operations was reduced by 15% by the cache management software.

This metric is only defined for internal disk units and is calculated as:

100 - ((DSDWOP/DIVISOR) * 100) / DSWRTS

DIVISOR depends on the disk configuration. It is 1 in non-RAID configurations, 2 for RAID 5, and 3 for RAID 6. It reflects the additional disk writes that result from the RAID.

In general, this metric is more important to the storage adapter cache management software designers. But it can also give an insight into workload behavior.

One has to be very careful with this metric. If RAID configuration is in the exposed mode due to a single disk failure, the number of physical disk writes will no longer correspond to the assumptions and this metric will no longer be valid.

Operation rate for read/write operations

One should watch for the rate of system disk operations. Some of the disk performance metrics are obtained by way of relatively slow sampling. When the operation rate is low, some calculated metrics, such as disk busy percentage, may become unstable.

Note: Here we are talking about disk operations sent by the system; this is different from disk device operations discussed before.

Read and write disk operations have very different behavior in terms of performance, so it is important to look at read and write operations separately, in addition to the total operation rate. The mixture of read/write operations also gives a clue about application behavior.

The operation rate per second is:

- DSRDS / INTSEC for read operations
- DSWRTS / INTSEC for write operations

Average operation size

It is more efficient to issue fewer disk operations that transfer more data than a large number of very small operations. This is why it is important to keep an eye on average operation size, that is, how much data is transferred by an average operation.

The operation size in kilobytes can be calculated as:

- For reads: (DSBLKR * 520) / DSRDS / 1024
- For writes: (DSBLKW * 520) / DSWRTS / 1024

These metrics are important for understanding the application's behavior. There are ways to improve these ratios by making sure that applications use blocking when appropriate, by changing logical page sizes of access paths used by application, and so on.

Note: When Expert Cache is turned on for one or more main storage pools, you may notice that this tends to increase the average size of disk operations. This is one of the ways used by the Expert Cache to improve performance.

The queries

In this section, an SQL query is shown that illustrates the some of the metrics discussed in this article. There are several assumptions that you must know before you start using these queries:

- The query should work in most cases, but it is not production quality code. It is provided *as-is*.
- This is a large query. It is too big to be comfortably used with a command-line tool such as STRSQL. We suggest using this query with the "Run SQL Scripts" tool available in iSeries Navigator. Make sure that option "Translate CCSID 65535" is chosen in the "JDBC Setup" settings.
- Comments were added to the query to make it easier to use and understand.
- This query demonstrates another technique for working with multi-member files. An SQL alias is created for a specific member and the query refers to the alias instead of the original file.
- All numerical fields are converted to floating point at the point of origin. This way, we do not have to worry about overflow and precision of intermediate results, which results in a better performance. It may, though, require reformatting of the fields when the end result is displayed.

Example A-3 shows the query.

Example A-3 SQL query (IBM i V5.4)

```
-- Create SQL alias for the member of QAPMDISK file.
create alias apytel.dsk for apytel.qapmdisk(Q154000003);

-- Set up default schema for name resolution.
set current schema apytel;


-- DISK DETAILS BY DISK UNIT BY INTERVAL
```

```
-- CTE for UNITDATA defines a table which brings all disk metrics to a disk unit
level.
-- This table can be used to construct higher-level queries.

with UNITDATA ( INTNUM, DTETIM, INTSEC, UNITNAME, DISKPOOL,
                AVGRSPT, PCTDSKBUSY, RDPERSEC, WRTPERSEC,
                AVGRDKB, AVGWRTKB, AVGSRVT, AVGWTT,
                PCTWRTOVER, PCTRDHIT, WRTEFC,
                DEVOPSEC,
                BK1RSPT, BK2RSPT, BK3RSPT, BK4RSPT,
                BK5RSPT, BK6RSPT )
as (

select

-- Interval number, timestamp and duration.
INTNUM, DTETIM, INTSEC,

-- Unit and disk pool identification.
DSARM || DMFLAG as UNITNAME,
DSASP || ' ' || DSASPN as DISKPOOL,

-- Average response time.
case when NUMCMD = 0 then 0 else
     case when RSPTNEW <> 0 then RSPTNEW/NUMCMDB
          else RSPTOLD/NUMCMD end
end as AVGRSPT,

-- Percent disk busy.
PCTDSKBUSY,

-- Read operations per second.
case when INTSEC = 0 then 0
     else NUMRD/INTSEC end as RDPERSEC,

-- Write operations per second.
case when INTSEC = 0 then 0
     else NUMWRT/INTSEC end as WRTPERSEC,

-- Average kilobytes per read operation.
case when NUMRD = 0 then 0
     else TOTKBR / NUMRD end as AVGRDKB,

-- Average kilobytes per write operation.
case when NUMWRT = 0 then 0
     else TOTKBW / NUMWRT end as AVGWRTKB,

-- Average Service time.
case when NUMCMD = 0 then 0 else
     case when SRVTNEW <> 0 then SRVTNEW/NUMCMDB
          else SRVTOLD/NUMCMD end
end as AVGSRVT,

-- Average wait time.
```

```
case when NUMCMD = 0 then 0 else
     case when SRVTNEW <> 0 then WTNEW/NUMCMDB
          else (RSPTOLD - SRVTOLD) / NUMCMD end
end as AVGWTT,

-- Write cache overrun %.
case when NUMWRT = 0 or NUMWRT <= FASTWRT then 0
     else ((NUMWRT - FASTWRT) * 100) / NUMWRT end as PCTWRTOVER,

-- Read cache hit %.
case when NUMRD = 0 or NUMRD <= RDHIT then 100
     else (RDHIT * 100) / NUMRD end as PCTRDHIT,

-- Write cache efficiency %.
case when NUMWRT = 0 then
       case when NUMDWRT = 0 then 100
            else 0 end
     else
     case when NUMWRT <= NUMDWRT/DIVISOR then 0
          else ((NUMWRT - (NUMDWRT/DIVISOR)) * 100) / NUMWRT
     end
end as WRTEFC,

-- Total device operation rate.
case when INTSEC = 0 then 0
     else (NUMDWRT + NUMDRD)/INTSEC end as DEVOPSEC,

-- Bucket 1:  RSPT < 1 ms.
case when DSBK1CMDS <> 0 then DSBK1RTTOT/DSBK1CMDS
     else 0 end as BK1RSPT,

-- Bucket 2:  1 ms <= RT < 16 ms.
case when DSBK2CMDS <> 0 then DSBK2RTTOT/DSBK2CMDS
     else 0 end as BK2RSPT,

-- Bucket 3:  16 ms <= RT < 64 ms.
case when DSBK3CMDS <> 0 then DSBK3RTTOT/DSBK3CMDS
     else 0 end as BK3RSPT,

-- Bucket 4:  64 ms <= RT < 256 ms.
case when DSBK4CMDS <> 0 then DSBK4RTTOT/DSBK4CMDS
     else 0 end as BK4RSPT,

-- Bucket 5:  256 ms <= RT < 1024 ms.
case when DSBK5CMDS <> 0 then DSBK5RTTOT/DSBK5CMDS
     else 0 end as BK5RSPT,

-- Bucket 6:  RT >= 1024 ms.
case when DSBK6CMDS <> 0 then DSBK6RTTOT/DSBK6CMDS
     else 0 end as BK6RSPT

from
(

-- Internal SELECT is used for convenience.
```

```
select

-- Information about collection interval.
INTNUM, min(DTETIM) as DTETIM, max(INTSEC) as INTSEC,

-- Disk unit and disk pool identification.
DSARM, DMFLAG, DSASP, DSASPN,

-- System disk operations.
sum(double(DSWRTS + DSRDS)) as NUMCMD,
sum(double(DSRDS)) as NUMRD,
sum(double(DSWRTS)) as NUMWRT,

-- Data for response time calculations.
sum((double(DSQUEL)/double(DSSMPL)) * double(INTSEC)) * 1000 AS RSPTOLD,
sum(double(DSSRVT + DSWT)) as RSPTNEW,

-- Disk busy percentage.
avg(double(DSSMPL - DSNBSY) / double(DSSMPL)) * 100 AS PCTDSKBUSY,

-- Data transferred.
sum(double(DSBLKR) * 520 / 1024 ) as TOTKBR,
sum(double(DSBLKW) * 520 / 1024 ) as TOTKBW,

-- Data for service and wait times calculation.
sum((double(DSSMPL - DSNBSY) / double(DSSMPL)) * double(INTSEC)) * 1000 AS
SRVTOLD,
sum(double(DSSRVT)) as SRVTNEW,
sum(double(DSWT)) as WTNEW,

-- Fast writes - cannot be both set.
sum(double(DSDCFW + DSCCFW)) as FASTWRT,

-- Read cache hits for external and internal disk units.
sum(case when binary(land(dscat,binary(x'01'))) = binary(x'01')
         then double(DSDCRH) else double(DSCCRH)
    end) as RDHIT,

-- Divisor for write efficiency.
case when DSPS = '1' then
     case when DSRDT = '0' then 2 else 3 end
     else 1
end as DIVISOR,

-- Disk device operations.
sum(double(DSDWOP)) as NUMDWRT,
sum(double(DSDROP)) as NUMDRD,

-- Disk response time buckets data.
sum(double(DSBKCT1)) as DSBK1CMDS,
sum(double(DSBKRT1)) as DSBK1RTTOT,
sum(double(DSBKCT2)) as DSBK2CMDS,
sum(double(DSBKRT2)) as DSBK2RTTOT,
sum(double(DSBKCT3)) as DSBK3CMDS,
```

```
sum(double(DSBKRT3)) as DSBK3RTTOT,
sum(double(DSBKCT4)) as DSBK4CMDS,
sum(double(DSBKRT4)) as DSBK4RTTOT,
sum(double(DSBKCT5)) as DSBK5CMDS,
sum(double(DSBKRT5)) as DSBK5RTTOT,
sum(double(DSBKCT6)) as DSBK6CMDS,
sum(double(DSBKRT6)) as DSBK6RTTOT,
sum(double(DSBKCT1+DSBKCT2+DSBKCT3+DSBKCT4+DSBKCT5+DSBKCT6)) as NUMCMDB

-- Here the alias is used.
from DSK

-- Aggregate data up to a disk unit per interval
GROUP BY INTNUM, DSARM, DMFLAG,
         DSASP, DSASPN, DSPS, DSRDT

)  T )

-- Now UNITDATA table can be used for other types of queries.
-- This example just displays all available data.

select *
from UNITDATA
order by INTNUM, UNITNAME;
```

B

A partition as a business

This appendix explains work management. To make grasping the overall concept of work management easier, we try comparing your IT system with a business.

© Copyright IBM Corp. 2009. All rights reserved.

Explaining your partition as a business

A simple partition can be compared to a small business, and a complex partition setup can be compared to a shopping mall. Assume that there is a small store in the business of building hand-crafted wood furniture. Work enters, such as orders for small tables, chairs, and bookshelves. Work is processed, that is, the carpenter calls the customers to confirm the order, and they are consulted on design points, including style, size, and color. The carpenter designs each piece of furniture, gathers the necessary materials, and then builds the furniture. After the furniture is completed, it is delivered (work leaves).

Since a complex system is a combination of many simple systems, a comparable example of a complex system is a shopping mall, many small and large businesses in one area. Maybe the carpenter has a business in the northwest corner of the mall and a baker has a business along the east strip. The baker and the carpenter have different inputs and outputs, that is, their orders and their products are very different. In addition, the time it takes each business to process their work is quite different, and their users know and understand that.

A complex partition (shopping mall) is a compilation of many smaller entities (stores). These entities are called subsystems.

Any piece of work within the business is considered a job. An example of a piece of work might be a customer letter, a telephone call, an order, or nightly cleanup. The same can be said about the IBM i product. On the system, each job has a unique name.

A job description describes how to handle the work coming into the subsystem. Job descriptions contain pieces of information, such as user IDs, job queues, and routing data. Information in the job description might compare to descriptions of jobs in a small business.

What the subsystem looks like

Every store has blueprints or store plans. These plans are really just descriptions, in varying detail, of the physical set up of the business. Maybe the business has a store with two floors, five doors, three mailboxes, and two telephones. On your partition, a subsystem description contains all the information about the subsystem and how to process the work.

Where the work comes from

For the carpenter, the work comes from customer calls, from references, and from people that stop in. On your system, the work can come from many places. Examples include job queues, workstations, communications, autostart jobs, and prestart jobs.

Where the jobs find the space

Within the mall, each business (subsystem) has a certain amount of floor space. On the partition, memory pools allow you to control the main storage (or floor space) each subsystem (business) gets to do its work. The more floor space a store (subsystem) has, the more customers, or jobs, can fit in the store. Of course, you must pay attention to the arrangements of the shelves, too. If the shelves are too tightly spaced there is no room for customers. The same applies to the number of customers in a shop: too many people in a

shop prevents everyone from moving easily through the shop. This equals having a too high activity level setting in a pool.

How the work comes in

Customers that cannot find the store they need may find an information booth to help send them in the right direction. The same is true on your partition. Routing entries are similar to store directories or an information booth. After the routing entry is found, it guides the job to its correct place. However, the routing entry needs to be found first, which is done by using routing data. Routing data is what the job uses to find the right routing entry.

How the work is treated

A carpenter needs to set a priority on each job. The chair due at the end of the week should be done before the bookshelf that is due at the end of the month. On your partition, classes provide information about how the job is handled while in the subsystem. This information includes priority while running, maximum storage, maximum CPU time, and time slice. Each of these attributes contribute to how and when a job is processed.

Just as there are rules that affect all the stores in the mall, there are rules that affect all the subsystems on your partition. An example of these rules is a system value. System values are pieces of information that apply to the whole system. System values include information such as date and time, configuration information, sign-on information, system security, and storage handling.

Customers in a mall each have information specific to them. On your partition, the user profile holds information specific to a particular user. Similar to a customer’s credit card, a user profile gives that user specific authorities and assigns the user attributes for that user's jobs. These job attributes provide information that includes, but is not limited to, the job description, the output queue or printer device, the message queue, the accounting code, and the scheduling priority.

Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 271. Note that some of the documents referenced here may be available in softcopy only.

- *Application and Program Performance Analysis Using PEX Statistics on IBM i5/OS*, SG24-7457
- *A Systems Management Guide to Performance Management for System i and System p servers*, SG24-7122
- *IBM iDoctor iSeries Job Watcher: Advanced Performance Tool*, SG24-6474

Online resources

These Web sites are also relevant as further information sources:

- Performance Management on IBM i

  http://www-03.ibm.com/systems/i/advantages/perfmgmt/
- IBM iDoctor for IBM i

  https://www-912.ibm.com/I_dir/idoctor.nsf
- Performance Management for Power Systems

  http://www-03.ibm.com/systems/power/support/pm/index.html
- Sizing tools on IBM i

  http://www-03.ibm.com/systems/i/advantages/perfmgmt/sizing.html
- Information Center for IBM i

  http://publib.boulder.ibm.com/iseries/

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this Web site:

ibm.com/redbooks

© Copyright IBM Corp. 2009. All rights reserved.

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A
action buttons 172
Actions on a collection 74
activity level 8
Add Job Watcher Definition 61
Add PEX Filter 69
Add Workload link 203
ADDPEXDFN) 69
ADDPEXFTR 69
advanced customization options 56
Advanced Growth Options button 191
agents
    IBM i 162
analyzing 4
ANZPFRDTA 23, 187
availability 6
average response time 6

B
baseline 10
batch throughput 144
batch work 11
blocked waits 32

C
capacity planning 4, 20
CIM 158
collecting 4
Collection Library 49
collection objects 21
Collection Retention Period 49
Collection Services 15–16
Command Language 17
Common Information Model 158
communications jobs 31
components 159
Computational Intensive Workloads 11
Convert Performance Collection 78
Copy Collection 75
CPU cycles 8
CPU history 175
CPU utilization 17
CPW 11
CPYPFRCOL 185–186
CPYPFRDTA 185
CRTGPHFMT 187
CRTPFRDTA 25, 48, 58, 186
Custom Profile 55
CVTPFRCOL 78
Cycle Interval 49
Cycle Time 49

D
data mining 29
Database Record 104
Delete Collection 76
Disk Arm Utilization 177
disk capacity 11
disk performance 242
disk reads 33
Disk Space Use history 177
Disk Statistical Overview 123
Disk Watcher 16, 23
Disk Watcher collector 36
disk writes 33
DMTF 158
drill-down 81
DSPPFRDTA 23
DSPPFRGPH 185, 187

E
elapsed time 6
ENDPEX 69
Enhanced Capacity Planning collection profile 55

F
faulting 11
faulting and paging 11
file monitors 130
filter 72

G
Graph Data Creation 50
Graph Data Retention 50
Graph History 179
Graph History function 22

H
handle 184
Health Indicators 38–39, 129
heterogeneous environments 158
Historical Data Creation 50
Historical Data Retention 50
HTTP server 31

I
I/O queuing 24
i5/OS 163
IBM Director 129
IBM i
    agents 162
IBM Systems Director Navigator for i 15
Ineligible 8

© Copyright IBM Corp. 2009. All rights reserved.

Input Line Time 8
interactive throughput 144
interactive transactions 6
interactive work 10
Investigate Data 76
IOPless storage adapters 252

J
Java Virtual Machine 31
job monitors 130
Job Watcher 16, 23
Job Watcher collector 36
Journal Management 164

K
Key Performance Indicator 227
KPI 227

L
logging in 160

M
machine pool faulting 222
main storage 8
Management Central 50
Management Central monitors 129
message monitors 130
Minimum collection profile 55
monitor types 130
Monitoring Agent for IBM i 165
multi-path support 252
multi-tiered applications 6

N
network performance 10

O
Object Lock 8, 104
object lock contention 8

P
paging 11
paging and faulting 11
partition 3
partition performance 17
Performance Data Investigator 48
Performance Explorer 16, 24
Performance Explorer collector 37
performance management 3
performance objectives 6
Performance Tools 15–16
Performance Tools plug-in 23
perspective 193
PEX Definition Types 68
planning
    upward integration 164
PM URL 170
Predefined Collection Profiles 54
Profile 68
PRTPEXRPT 39

Q
QAPEXDFN file 69
QBASACTLV 216
QBASPOOL 216
QCMNARB 216
QCMNRCYLMT 217
QDBFSTCCOL 217
QDYNPTYADJ 217
QDYNPTYSCD 217
QLIBLCKLVL 217
QMAXACTLVL 217
QMCHPOOL 217
QPASTHRSVR 217
QPFRADJ 146, 216–217
QPRCMLTTSK 217
QQRYDEGREE 217
QQRYTIMLMT 217
QTHDRSCADJ 218
QTHDRSCAFN 218
QTSEPOOL 218
Queued for the CPU state 32

R
read cache 248
real-time status 150
record locks 33
Redbooks Web site 271
    Contact us xiii
response time 6
response time, perceived 6
Restore Performance Collection 78
Retired Sizing 211
RSTPFRCOL 78

S
Seize contention 104
Select Group button 171
Selection and Navigation 73
SEQ parameter 186
Short Wait 8
Show Tool Tips 91
SLA 6
Sort 73
spool throughput 144
SQL Performance Monitors 22
SQL Script Center 22
SQL Visual Explain 22
Standard collection profile 55
Standard Plus Protocol collection profile 55
Statistical collection type 68
STRDBMON 22
STRPEX 69
STRPFRCOL 25

System i Navigation Monitors 15
System i Navigator Monitors 16
system monitors 130
System Resources 156
System Response Time 8

T

threshold commands 134
thresholds 134
throughput 6
Trace collection type 68
trend analysis 170

U

upward integration 164
user expectations 5
User Response Time 8
user-defined transactions 57

V

Visual Explain 22

W

Wait - blocked state 32
Wait - Idle state 32
wait accounting 29
wait buckets 244
Wait state 242
Watch for Event function 129
Web interface
    starting 160
    welcome page 161
welcome page 161
WLE 21
work management 9
Workload definition 190
Workload Estimator 20, 40
Workload selection 204
Write cache 249
WRKACTJOB 23, 25, 186
WRKDSKSTS 18, 25, 186
WRKPRB 18
WRKPTFGRP 18
WRKSYSACT 23, 25, 186
WRKSYSSTS 11, 23, 25, 186

End to End Performance Management on IBM i

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

End to End Performance Management on IBM i

Understand the cycle of Performance Management

Maximize performance using the new graphical interface on V6.1

Learn tips and best practices

Monitoring and managing your system's performance is critical to ensure that you are keeping pace with the changing demands of your business.

To respond to business changes effectively, your system must change too. Managing your system, at first glance, might seem like just another time-consuming job. But the investment soon pays off because the system runs more efficiently, and this is reflected in your business. It is efficient because changes are planned and managed.

Managing performance of any system can be a complex task that requires a thorough understanding of that system's hardware and software. IBM i is an industry leader in the area of performance management and has many qualities that are not found in other systems, such as:

- Unparalleled performance metrics
- Always-on collection of metrics
- Graphical investigation of performance data

While understanding all the different processes that affect system performance can be challenging and resolving performance problems requires the effective use of a large suite of tools, the functions offered by IBM i are intended to make this job easier for users.

This IBM Redbooks publication explains the tasks and rich tools associated with performance management on IBM i.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7808-00 ISBN 0738433810