

# IBM i on Power - Performance FAQ

*May 7, 2024*

## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>12</b>
1.1	PURPOSE OF THIS DOCUMENT .....	12
1.2	OVERVIEW.....	12
1.3	WHAT’S NEW IN THE LATEST VERSION?.....	12
1.4	DOCUMENT RESPONSIBILITIES .....	12
<b>2</b>	<b>WHAT IS PERFORMANCE?.....</b>	<b>13</b>
2.1	INTRODUCTION .....	13
2.2	PERFORMANCE .....	13
2.3	RESPONSE TIME.....	13
2.4	THROUGHPUT.....	13
2.5	THROUGHPUT VERSUS RESPONSE TIME .....	13
2.6	COMPONENTS OF RESPONSE TIME.....	14
2.6.1	<i>Transaction Characteristics</i> .....	15
2.6.2	<i>IBM i Wait Buckets</i> .....	16
2.6.3	<i> HOLDERS and Waiters</i> .....	17
2.6.4	<i>Call Stacks</i> .....	17
2.6.5	<i>Collecting and Analyzing the Data</i> .....	18
2.7	ACCEPTABLE PERFORMANCE .....	18
2.8	PARTITION SCALING CONSIDERATIONS.....	18
<b>3</b>	<b>PERFORMANCE BENCHMARKS.....</b>	<b>20</b>
3.1	INTRODUCTION .....	20
3.2	WHAT ARE PERFORMANCE BENCHMARKS?.....	20
3.3	HOW ARE IBM I SYSTEMS RATED FOR PERFORMANCE? .....	20
3.4	WHAT IS THE CPW RATING OF A SYSTEM?.....	20
3.4.1	<i>CPWRating vs. Public Benchmarks</i> .....	21
3.5	COMPARING BENCHMARK RESULTS .....	21
3.6	CUSTOM BENCHMARKS AND PROOF OF CONCEPTS (POC).....	21
3.6.1	<i>Considerations</i> .....	22
3.6.2	<i>Common Pitfalls for Custom Benchmarks</i> .....	22
3.6.3	<i>IBM i Performance &amp; Scalability Services Center</i> .....	22
<b>4</b>	<b>SIZING A SYSTEM.....</b>	<b>23</b>
4.1	INTRODUCTION .....	23
4.2	SIZING BEST PRACTICES.....	23
4.3	SYSTEM SIZING TOOLS.....	23
4.3.1	<i>IBM Systems Workload Estimator (WLE)</i> .....	23
4.3.2	<i>IBM Storage Modeller</i> .....	25
<b>5</b>	<b>PROACTIVE PERFORMANCE MONITORING .....</b>	<b>26</b>
5.1	INTRODUCTION .....	26
5.2	PERFORMANCE MONITORING BEFORE A PROBLEM OCCURS .....	26
5.2.1	<i>Real-time Interactive Monitoring</i> .....	26
5.2.2	<i>Dashboard on Navigator for i</i> .....	26
5.2.3	<i>Automated Monitoring</i> .....	28
5.2.4	<i>Watching for messages</i> .....	29

5.2.5	IBM i Services.....	29
5.3	DEFINING A PERFORMANCE PROBLEM (WHAT IS SLOW?).....	29
<b>6</b>	<b>PERFORMANCE DATA COLLECTORS AND ANALYSIS TOOLS.....</b>	<b>30</b>
6.1	INTRODUCTION.....	30
6.2	MANAGING PERFORMANCE ON IBM I.....	30
6.3	PERFORMANCE DATA COLLECTORS.....	30
6.3.1	Collection Services.....	30
6.3.2	IBM i Disk Watcher.....	31
6.3.3	IBM i Job Watcher.....	32
6.3.4	Performance Explorer (PEX).....	32
6.3.5	SQL Plan Cache.....	33
6.4	ANALYSIS TOOLS.....	33
6.4.1	Overview.....	33
6.4.2	Performance Tools for i.....	34
6.4.3	IBM i Access Client Solutions (ACS).....	34
6.4.4	IBM Navigator for i.....	34
6.4.5	Performance Data Investigator.....	35
6.4.6	IBM iDoctor for IBM i.....	37
6.4.7	Tools Matrix.....	39
6.4.8	Open Source Languages.....	39
<b>7</b>	<b>FREQUENTLY ASKED QUESTIONS.....</b>	<b>40</b>
7.1	INTRODUCTION.....	40
7.2	GENERAL QUESTIONS.....	40
7.2.1	What are the general performance tuning recommendations for best IBM i performance?.....	40
7.2.2	I heard that... something on the internet said... should I change...?.....	40
7.2.3	My <CPU, disk, memory> usage is high, should I add more resource?.....	40
7.2.4	Why didn't the performance of my partition improve after I added <CPU, memory, disk arms, SSDs>? 40	40
7.2.5	Should I update PTFs before undertaking a performance analysis project?.....	40
7.2.6	How can I find out which jobs are consuming the most temporary storage on my partition?.....	41
7.2.7	There are two graphical interfaces for IBM i performance data – Performance Data Investigator (PDI) and iDoctor. Which should I use?.....	42
7.2.8	What is the job QDBFSTCCOL running on my partition and why is it taking resources?.....	43
7.2.9	How can I better isolate PASE jobs for analysis?.....	44
7.3	CPU QUESTIONS.....	44
7.3.1	How can I reduce CPU usage on the partition?.....	44
7.3.2	What can I do if I can't reduce my CPU consumption?.....	44
7.3.3	Why does my partition's CPU utilization show a value greater than 100%?.....	44
7.3.4	Why doesn't the System CPU utilization equal the sum of the CPU utilization of the individual jobs in WRKACTJOB?.....	45
7.3.5	How can I keep certain workloads from impacting my whole partition?.....	45
7.3.6	The performance of my system is inconsistent. What could cause that?.....	45
7.3.7	Why do I see CPU queuing time when my CPU utilization is not excessive?.....	46
7.3.8	Why didn't my performance improve when I added cores to my partition?.....	46
7.3.9	Why doesn't the CPU utilization reported by the HMC match the CPU utilization reported by IBM i?... 47	47
7.4	MEMORY QUESTIONS.....	47
7.4.1	How much memory should I configure for my partition?.....	47
7.4.2	How should I configure my partition's memory pools?.....	47

7.4.3	<i>Should I turn on the Performance Adjuster (QPFRADJ)?</i> .....	47
7.4.4	<i>How can I tell what is in my partition's memory pools?</i> .....	47
7.4.5	<i>When should I add more memory to my partition?</i> .....	48
7.4.6	<i>What are the "Memory page fault guidelines"?</i> .....	48
7.5	I/O QUESTIONS.....	48
7.5.1	<i>What are some common storage subsystem configuration changes that can hurt performance?</i> .....	48
7.5.2	<i>How can I analyze I/O subsystem performance?</i> .....	48
7.5.3	<i>I can't modify my applications. What can I do to improve my I/O performance?</i> .....	48
7.5.4	<i>How do I virtualize IBM Serial-Attached SCSI (SAS) adapters for the best performance?</i> .....	49
7.5.5	<i>How many and what type of storage I/O products meet performance requirements for an upgrade or new system or workload?</i> .....	49
7.5.6	<i>How can I tell if a FlashSystem solution is the best storage configuration for my environment?</i> .....	49
7.5.7	<i>How can I tune my Virtual Fibre Channel attached external storage?</i> .....	50
7.5.8	<i>How do I analyze I/O performance if I'm using external storage?</i> .....	50
7.5.9	<i>What are the IBM i performance effects of using 4096 byte sector drives?</i> .....	51
7.5.10	<i>Should I utilize NVMe drives?</i> .....	51
7.6	PROCESSOR VIRTUALIZATION QUESTIONS.....	52
7.6.1	<i>How do I determine if my workload is being impacted by processing capacity entitlement delay?</i> .....	52
7.6.2	<i>How can processing capacity entitlement delays be reduced?</i> .....	52
7.6.3	<i>How do I determine if my workload is being impacted by over-commit of the shared processor pool?</i> 54	
7.6.4	<i>How can virtual processor dispatch latency delays be reduced?</i> .....	54
7.7	HOW DO I TUNE IBM I DATABASE PERFORMANCE? .....	56
7.8	HOW CAN I IMPROVE BACKUP AND RECOVERY TIMES?.....	56
7.9	HOW CAN I IMPROVE IPL TIME?.....	56
7.10	WHAT ARE CONSIDERATIONS FOR MAIN STORE DUMP (MSD) IPL PERFORMANCE? .....	56
7.11	HOW DO I TUNE JOURNAL AND RECOVERY PERFORMANCE? .....	56
7.12	HOW DO I TUNE ACCESS PATH RECOVERY PERFORMANCE? .....	57
7.13	HOW DO I TUNE NETWORK PERFORMANCE? .....	58
7.14	HOW CAN I IMPROVE THE PERFORMANCE OF JAVA AND/OR WEBSHERE APPLICATION SERVER APPLICATIONS? .....	59
7.15	HOW DO I TUNE RPG/COBOL APPLICATION PERFORMANCE AND NATIVE I/O FILE ACCESS?.....	59
7.16	CAN SECURITY SETTINGS AFFECT PERFORMANCE? .....	59
7.17	WHAT ARE SOME INEFFICIENT APPLICATION ISSUES I CAN IDENTIFY AT A SYSTEM LEVEL? .....	60
7.17.1	<i>Look for high rates of database full opens</i> .....	60
7.17.2	<i>Look for high rates of activation group creates/destroys</i> .....	61
7.18	WHAT IS THE LARGEST NUMBER OF CORES THAT CAN BE ASSIGNED TO AN IBM I PARTITION?.....	61
7.19	HOW CAN I IMPROVE THE SCALABILITY OF MY LARGE IBM I PARTITION?.....	62
7.20	HOW CAN I PROACTIVELY ENSURE THAT BATCH PERFORMANCE WILL IMPROVE WITH MY SYSTEM UPGRADE?.....	63
7.21	HOW CAN I CHECK THE "INTERACTIVE CAPACITY" SETTING ON MY SYSTEM?.....	63
7.22	WHAT ARE CONSIDERATIONS FOR IBM POWER VIRTUAL SERVER PARTITIONS.....	64
7.23	WHAT ARE SOME CONSIDERATIONS FOR USE OF DATA COMPRESSION IN IBM I? .....	65
7.24	SHOULD I UTILIZE ASP ENCRYPTION?.....	67
7.25	WHERE CAN I FIND MORE INFORMATION ON PERFORMANCE TUNING? .....	68
7.26	IBM TECHNOLOGY EXPERT LABS CAN HELP .....	68
<b>8</b>	<b>DATABASE PERFORMANCE</b> .....	<b>69</b>
8.1	ARE THERE SOME BASIC SUGGESTIONS FOR OPTIMAL SQL PERFORMANCE? .....	69
8.2	HOW CAN I IMPROVE PERFORMANCE OF APPLICATIONS THAT USE RECORD LEVEL ACCESS (NATIVE I/O)?.....	70
8.3	WHAT TOOLS ARE AVAILABLE TO MONITOR AND TUNE SQL? .....	70
8.4	IBMDB2 MIRROR FOR I .....	77

8.5	MORE INFORMATION ON Db2 FOR I PERFORMANCE AND SUPPORT .....	78
<b>9</b>	<b>IBM I SERVICES.....</b>	<b>79</b>
9.1	INTRODUCTION .....	79
9.2	IBM I SERVICES CURRENTLY AVAILABLE .....	79
9.3	Db2 FOR I SERVICES CURRENTLY AVAILABLE .....	80
9.4	IBM I ACCESS CLIENT SOLUTIONS.....	80
9.5	FOR MORE INFORMATION: .....	82
<b>10</b>	<b>RPG/COBOL AND NATIVE I/O .....</b>	<b>83</b>
10.1	INTRODUCTION .....	83
10.2	CPU PERFORMANCE TIPS.....	83
10.2.1	<i>Compiler Options</i> .....	83
10.2.2	<i>Profiling</i> .....	83
10.3	IO PERFORMANCE TIPS .....	84
10.3.1	<i>General tips</i> .....	84
10.3.2	<i>Database modernization</i> .....	84
<b>11</b>	<b>C PROGRAM COMPILER OPTIMIZATION .....</b>	<b>85</b>
11.1	IBM I ARCHITECTURE – MI INTERFACE.....	85
11.2	IBM I PROGRAM CREATION .....	86
11.3	COMPILER OPTION - OPTIMIZE.....	86
11.4	PROGRAM PROFILING .....	87
11.5	ADVANCED ARGUMENT OPTIMIZATION.....	88
11.6	INTERPROCEDURAL ANALYSIS (IPA).....	89
<b>12</b>	<b>JAVA AND WEBSHERE .....</b>	<b>90</b>
12.1	INTRODUCTION .....	90
12.2	DOES IT MATTER IF I USE THE 32-BIT VERSUS 64-BIT JVM?.....	90
12.3	I'M USING AN OLD JDK LEVEL AND AM BEHIND ON PTFs, IS THAT OKAY? .....	90
12.4	SHOULD I RUN MY JVMs IN A SEPARATE MEMORY POOL?.....	90
12.5	HOW DO I DETERMINE THE PROPER GC SETTINGS (HEAP SIZES, COLLECTION POLICY, ETC.) FOR MY APPLICATION? .....	90
12.6	HOW DO I ANALYZE JVM DUMPS AND VERBOSE GC OUTPUT?.....	91
12.7	HOW DO I TUNE JAVA CODE?.....	91
12.8	WHERE CAN I GET MORE INFORMATION ON JAVA TUNING AND DEBUGGING?.....	91
12.9	CAN I USE JOB WATCHER TO COLLECT JAVA INFORMATION? .....	91
12.10	HOW CAN I IMPROVE THE PERFORMANCE OF WEBSHERE APPLICATION SERVER RUNNING ON IBM I?.....	91
12.11	HOW DO I TUNE IBM TOOLBOX FOR JAVA PERFORMANCE? .....	91
12.12	WHERE CAN I FIND ADDITIONAL INFORMATION?.....	92
<b>13</b>	<b>POWER PROCESSOR-BASED SYSTEMS .....</b>	<b>93</b>
13.1	INTRODUCTION .....	93
13.2	SIMULTANEOUS MULTITHREADING (SMT) AND INTELLIGENT THREADS.....	93
13.2.1	<i>Single thread (ST) versus SMT2/SMT4/SMT8</i> .....	93
13.2.2	<i>Intelligent Threads Technology</i> .....	93
13.3	FLEXIBLE SMT CONTROLS .....	94
13.3.1	<i>Processor Compatibility Mode</i> .....	94
13.3.2	<i>QPRCMLTTSK System Value</i> .....	95
13.3.3	<i>QWCCHGPR API</i> .....	96
13.4	POWER SAVINGS MODE.....	96

- 13.5 ADAPTER PLACEMENT..... 97
- 13.6 AFFINITIZED PARTITIONS..... 97
- 13.7 SYSTEM AFFINITY SETTINGS..... 98
- 13.8 COLLECTING AND VIEWING PERFORMANCE DATA FOR ENTIRE PHYSICAL SYSTEM..... 98
  - 13.8.1 *Viewing Performance Data for Multiple Systems*..... 102
- 13.9 HOW DOES IBM I REPORT CPU UTILIZATION FOR SMT PROCESSORS?..... 103
- 13.10 HOW IS IBM I CPU UTILIZATION CALIBRATED FOR SMT PROCESSORS?..... 104
- 13.11 POWER8 REDBOOK..... 105
- 13.12 POWER9 AND POWER10 REDBOOKS ..... 105
- 14 VIRTUALIZATION..... 106**
  - 14.1 VIRTUALIZATION BEST PRACTICES ..... 106
  - 14.2 DYNAMIC PLATFORM OPTIMIZER (DPO)..... 106
  - 14.3 VIOS ADVISOR ..... 106
- 15 REPORTING A PERFORMANCE PROBLEM..... 108**
  - 15.1 INTRODUCTION ..... 108
  - 15.2 DEFINE THE PERFORMANCE PROBLEM ..... 108
  - 15.3 QUESTIONS THAT HELP IBM DIAGNOSE THE PROBLEM..... 108
- 16 REFERENCES..... 110**
  - 16.1 BLOGS ON PERFORMANCE TOPICS ..... 110
  - 16.2 EDUCATION AND TRAINING ..... 110
  - 16.3 REDBOOKS AND REDPAPERS..... 110
  - 16.4 ADDITIONAL PERFORMANCE-RELATED INFORMATION..... 111
  - 16.5 DATABASE PERFORMANCE ARTICLES ..... 111
  - 16.6 KEY PERFORMANCE RESOURCES ..... 111
  - 16.7 PERFORMANCE CAPABILITIES REFERENCE NOTE..... 112

## Table of Figures

Figure 1 - IBM Navigator for i Dashboard .....	27
Figure 2 - IBM Navigator for i Dashboard Table View.....	27
Figure 3 - IBM Navigator for i System Monitors .....	28
Figure 4 - Collector data characteristics .....	33
Figure 5 - Performance Data Investigator .....	35
Figure 6 - Performance Data Investigator example.....	36
Figure 7 - Graph History example.....	37
Figure 8 - iDoctor components .....	38
Figure 9 - iDoctor example.....	38
Figure 10 - Relationship between collector and major functions / interfaces .....	39
Figure 11 - Active Jobs Temporary Storage Used - Navigator for i.....	41
Figure 12 – System Temporary Storage Consumption by Active Jobs - IBM i Services .....	42
Figure 13 – Temporary Storage Used - Navigator for i .....	42
Figure 14 - PDI and iDoctor feature comparison .....	43
Figure 15 - Partition-level illustration of entitlement delay reduction - IBM iDoctor for i. ....	53
Figure 16 - Job-level illustration of processor virtualization delay reduction - IBM iDoctor for i.....	53
Figure 17 - Partition-level illustration of processor virtualization delay - IBM iDoctor for i.....	55
Figure 18 - Job-level illustration of processor virtualization delay - IBM iDoctor for i.....	55
Figure 19 - Network Link Aggregation.....	59
Figure 20 - Full Opens in iDoctor Collection Services Investigator .....	60
Figure 21 - Activation Groups Created in iDoctor Collection Services Investigator .....	61
Figure 22 - Maximum Cores in Partition.....	62
Figure 23 - Interactive Capacity .....	64
Figure 24 - Compression Options .....	66
Figure 25 - Compression Times by Library Size.....	67
Figure 26 - SQL Performance Center.....	70
Figure 27 - SQL Plan Cache.....	71
Figure 28 - SQL Database Monitor.....	72
Figure 29- Visual Explain .....	73
Figure 30 - Index Advisor.....	73
Figure 31 - Index Evaluator .....	74
Figure 32 - PDI Database Package.....	74
Figure 33 - PDI Physical Database I/O by Job or Task.....	75
Figure 34 - PDI - Collection Services perspectives in Database Package .....	76
Figure 35 - Collection Services - Query Opens Chart.....	77
Figure 36 - IBM i Access Client Solutions - IBM i Services Examples.....	81
Figure 37 - IBM i Access Client Solutions – Active Job Info – Longest running SQL statements.....	81
Figure 38 – IBM i Architecture – MI Interface .....	85
Figure 39 - Processor Compatibility Mode.....	95

- Figure 40 - Partition Placement.....98
- Figure 41 - Partition Properties .....99
- Figure 42 - PDI Physical System Perspectives .....100
- Figure 43 - Logical Partitions Overview .....101
- Figure 44 - iDoctor System Graphs.....101
- Figure 45 - Prometheus visualization with Grafana .....102
- Figure 46 - IBM i SMT8 CPU Utilization Reporting.....105



## Preface

This document is intended to address most frequently asked questions concerning IBM i performance on Power and provide best practice guidelines for most commonly seen performance issues.

The following is a list of key IBM® reference and documents. [The References section](#) contains additional references.

- The End to End Performance Management on IBM i Redbook provides a comprehensive set of performance tips and guidance with focus on monitoring and managing IBM i system performance: [End to End Performance Management on IBM i Redbook](#)
- The Performance Optimization and Tuning Techniques for IBM Power Systems Processors Including IBM Power8 Redbook provides performance information including specific tuning for Power8. [Perf Optimization & Tuning Techniques Including IBM POWER8 Redbook](#)
- IBM Documentation has a section on performance: [Performance](#)
- As well as this subsection on Managing System Performance: [Managing system performance](#)
- The following link provides a wide variety of information on topics for IBM i including performance: <https://developer.ibm.com/components/ibm-i/>
- You can also use the IBM Support resource to provide information on IBM i performance tools topics: [Performance Tools](#)

## Acknowledgments

We would like to thank the many people who made invaluable contributions to this document. Contributions included authoring, insights, ideas, reviews, critiques and reference documents

Our special thanks to key contributors from IBM Power Performance:

- Clark Anderson – I/O Performance
- Grace Bowser – IBM i Performance
- Dirk Michel – Power Performance
- Rick Peterson – IBM i Performance

Our special thanks to key contributors from IBM i Development Support:

- Colin Devilbiss – IBM i Development
- Scott Forstie – IBM i Development
- Chris Francois – IBM i Development
- Lora Powell – IBM i Development
- Shauna Rollings – IBM i Development

Our special thanks to key contributors from IBM i Global Support Center:

- Kevin Chidester - IBM i Performance
- Brad Menges – IBM i Performance

Our special thanks to key contributors from IBM Technology Expert Labs:

- Eric Barsness – IBM i Performance
- Stacy Benfield – IBM i Performance
- Terry Ford – IBM i Security
- Kent Milligan - Db2 for i Performance
- Paul Nelsestuen – IBM i Performance

## Disclaimer – Performance FAQ

Copyright © 2024 by International Business Machines Corporation.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information may include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.

Statements regarding IBM's future direction and intent are subject to change or withdrawal without notice and represent goals and objectives only.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

# 1 Introduction

## 1.1 Purpose of this document

The purpose of this document is to provide a basic understanding of IBM i on Power performance concepts, workloads and benchmarks on Power, capacity planning, performance monitoring and analysis, frequently asked questions and guidelines addressing common performance issues.

This document is not intended to replace performance management documentation or performance white papers.

## 1.2 Overview

This document covers a variety of Power performance including:

- What Is Performance?
- Performance Benchmarks
- Sizing a System and Capacity Planning
- Performance Tools
- Performance Analysis and Tuning
- Frequently Asked Questions
- Reporting a Performance Problem

## 1.3 What's new in the latest version?

Several new sections were added or changed in this version:

1. Frequently Asked Questions updates
2. Link updates throughout document

## 1.4 Document Responsibilities

The IBM Systems Power Performance organization is responsible for editing and maintaining the IBM i on Power - Performance FAQ document. Any contributions or suggestions for additions or edits should be forwarded to Stacy Benfield, [stacylb@us.ibm.com](mailto:stacylb@us.ibm.com).

## 2 What Is Performance?

### 2.1 Introduction

The purpose of this chapter is to explain what exactly computer performance is.

### 2.2 Performance

Computer performance is largely determined by a combination of response time and throughput. Other aspects associated with computer performance are availability of computer systems and their power efficiency.

### 2.3 Response Time

The response time of a computer system is the elapsed time between the end of a transaction inquiry or demand and the beginning of a response to that transaction. For interactive users, the response time is the time from when the user hits the <enter> button to seeing the result displayed. The response time often is seen as a critical aspect of performance because of its potential visibility to end users or customers.

Let's take the example of a computer system that is running a web server with an online store. The response time here is the elapsed time between pressing the submit button to place an order and the beginning of receiving the order confirmation.

### 2.4 Throughput

The throughput of a computer system is a measure of the amount of work performed by a computer system over the period of time. Examples for throughput are megabytes per second read from a disk, database transactions per minute, megabytes transmitted per second through a network adapter.

Let's go back to the previous example with the online store. The computer system on which the online store is running might be one out of many computers that are all connected to the same database server. While response time of the database server is an important factor, its throughput may be more important because it processes many requests from the web servers in parallel.

### 2.5 Throughput versus Response Time

Throughput and response time are related. As system resources (processors, memory, disk, etc.) are used by multiple, simultaneous transactions, there may be delays to individual transactions because other transactions are using those particular resources when they are demanded and the request for processing is temporarily delayed. System resources that are shared can cause transactions that rely on these resources to incur a response time increase due to this queuing.

However, Power processor-based systems are designed for high-performance sharing. As transaction load increases on the system, throughput increases. It follows that, as resource queuing increases, response time increases too. Therefore, high throughput may come at the cost of slower response times.

Let's assume you load a truck with 10,000 1TB disks, drive the truck 30 miles, and unload the disks

within one hour. The throughput would be 2.8TB per second but at the cost of the response time, which would be an hour.

Now take a sports car instead of a truck. It's unlikely that 10,000 disks would fit into a small car, so let's assume we can load 100 1TB disks, drive the sports car 30 miles, and unload the disks within 20 minutes. The response time now is three times better compared to the truck, however, the throughput reduced to 0.083TB per second.

### 2.6 Components of Response Time

Wait Accounting is the patented technology built into the IBM® i operating system that tells you what a thread or task is doing when it appears that it is not doing anything. When a thread or task is not executing, it is waiting. Wait accounting, a concept exclusive to IBM i, is a very powerful capability for detailed performance analysis. The following information is going to focus on waiting, why threads wait, and how you can use wait accounting to troubleshoot performance problems or to simply improve the performance of your applications.

A job is the basic mechanism through which work is done. Every job has at least one thread and may have multiple threads. Every thread is represented by a Licensed Internal Code (LIC) task, but tasks also exist without the IBM i thread-level structures. LIC tasks are generally not visible externally except through the IBM i Performance or Service Tools. Wait accounting concepts apply to both threads, and tasks, thus, the terms thread and task are used when referring to an executable piece of work.

A thread or task has two basic states it can be in:

- Executing on the processor. This is the "running" state.
- Waiting

There are three key wait conditions:

- Ready to run, waiting for the processor. This is a special wait state and is generally referred to as "*CPU Queuing*." This means the thread or task is queued and is waiting to run on the CPU. There are a few different reasons that CPU queuing can occur. An example could be if the partition is overloaded and there is more work than the partition can accommodate, then work can be queued to wait for the CPU. This can be compared to a highway that has ramp meters; when the highway is congested, the ramp meters have a red signal so that the cars have to stop and wait before they can enter traffic. Logical partitioning and simultaneous multithreading can also result in CPU queuing.
- Idle waits. Idle waits are a normal and expected wait condition. Idle waits occur when the thread is waiting for external input. This input may come from a user, the network, or another application. Until that input is received, there is no work to be done.
- Blocked waits. Blocked waits are a result of serialization mechanisms to synchronize access to shared resources. Blocked waits can be normal and expected. Examples include serialized access to updating a row in a table, disk I/O operations, or communications I/O operations. However, blocked waits may not be normal, and it is these unexpected block points that are situations where wait accounting can be used to analyze the wait conditions.

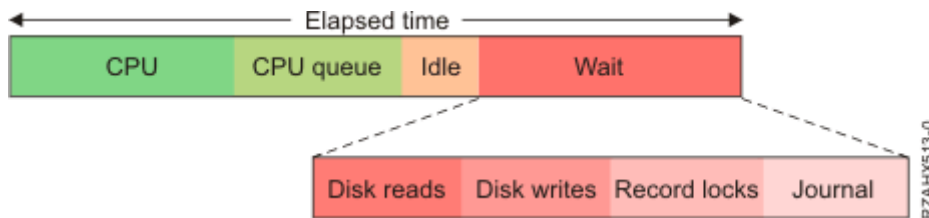
You can think of the lifetime of a thread or a task in a graphical manner, breaking out the time spent running or waiting. This graphical description is called the "run-wait time signature". At a high-level, this signature looks as follows (each of the colored segments represents a certain amount of cumulative time the job spent in that wait group during its run-time):



Traditionally, the focus for improving the performance of an application was to have it use the CPU as efficiently as possible. On IBM i with wait accounting, we can examine the time spent waiting and understand what contributed to that wait time. If there are elements of waiting that can be reduced or eliminated, then the overall performance can also be improved.

Nearly all of the wait conditions in the IBM i operating system are identified and enumerated - that is, each unique wait point is assigned a numerical value. This is possible because IBM has complete control over both the licensed internal code and the operating system. As of the IBM i 6.1 and 7.1 releases, there are 268 unique wait conditions. Keeping track of over 250 unique wait conditions for every thread and task would consume too much storage, so a grouping approach is used. Each unique wait condition is assigned to one of 32 groups, or "buckets". As threads or tasks go into and out of wait conditions, the task dispatcher maps the wait condition to the appropriate group.

If we take the run-wait time signature, that uses wait accounting, we can now identify the components that make up the time the thread or task was waiting. For example:



If the thread's wait time was due to reading and writing data to disk, locking records for serialized access, and journaling the data, we could see the waits broken out above. When you understand the types of waits that are involved, you can start to ask yourself some questions. For the example above, some of the questions that could be asked are:

- Are disk reads causing page faults? If so, are my pool sizes appropriate?
- What programs are causing the disk reads and writes? Is there unnecessary I/O that can be reduced or eliminated? Or can the I/O be done asynchronously?
- Is my record locking strategy optimal? Or am I locking records unnecessarily?
- What files are being journaled? Are all the journals required and optimally configured?

### 2.6.1 Transaction Characteristics

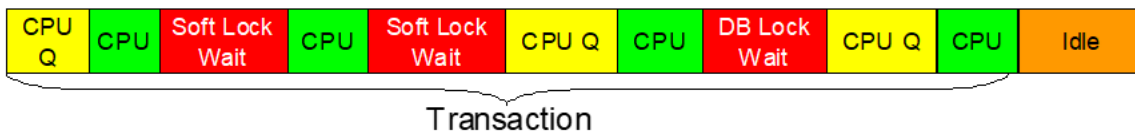
Transactions that consist of a mix of considerable disk wait times are called **I/O-limited transactions**. These transactions gain the most benefit from an efficient disk I/O subsystem. High-speed disks, large disk cache, and high-speed I/O links can improve the response time and throughput of these types of transactions. Also, consider SSD or NVMe as possible alternatives to HDDs for this type of transactions. See [I/O Questions](#) for more tips for improving I/O performance.

**I/O-limited Transaction:**



Transactions that consist of a considerable mix of waits and processor activity are called **resources-limited transactions**. These transactions tend to benefit most from improving application efficiency. Reduce the impact of critical shared resources between applications or program threads. Collection Services and Job Watcher data can be used to help determine the critical resources and waiters. Use Performance Data Investigator (PDI) or IBM iDoctor to analyze the data.

**Resources-limited transaction:**



Transactions that consume considerable processor cycles and very little I/O or wait times are called **processor-bound transactions**. These transactions tend to benefit most from higher processor frequency and the efficiency of the processor pipeline stages. They may also benefit from [improved processor and memory nodal affinity](#) and from improved partition placement (e.g. [Dynamic Platform Optimizer "DPO"](#)).

**Processor-bound transaction:**



**2.6.2 IBM i Wait Buckets**

The following 32 wait groups or "buckets" are defined. The definition of the wait groups varies from release to release and may change in the future.

1. Time dispatched on a CPU
2. CPU queuing
3. Reserved
4. Other waits



5. Disk page faults
6. Disk non-fault reads
7. Disk space usage contention
8. Disk operation start contention
9. Disk writes
10. Disk other
11. Journaling
12. Semaphore contention
13. Mutex contention
14. Machine level gate serialization
15. Seize contention
16. Database record lock contention
17. Object lock contention
18. Ineligible waits
19. Main storage pool contention
20. Journal save while active
21. Reserved
22. Reserved
23. Reserved
24. Socket transmits
25. Socket receives
26. Socket other
27. IFS
28. PASE
29. Data queue receives
30. Idle/waiting for work
31. Synchronization Token contention
32. Abnormal contention

There are many of these wait groups that you may see surface if you do wait analysis on your application. Understanding what your application is doing and why it is waiting in those situations can possibly help you reduce or eliminate unnecessary waits.

For more information on these wait groups, refer to these Job Waits Whitepapers:

[Job Waits Whitepaper](#)

or

[IBM i Wait Accounting](#)

### **2.6.3 Holders and Waiters**

Not only does IBM i keep track of what resource a thread or task is waiting on, it also keeps track of the thread or task that has the resource allocated to it. This is a very powerful feature. A "holder" is the thread or task that is using the serialized resource. A "waiter" is the thread or task that wants access to that serialized resource.

### **2.6.4 Call Stacks**

IBM i also manages call stacks for every thread or task. This is independent of the wait accounting

information. The call stack shows the programs that were invoked and can be useful in understanding the wait condition, such as knowing some of the logic that led up to either holding a resource or wanting to get access to it. The combination of holder, waiter, and call stacks provide a very powerful capability to analyze wait conditions. Beginning with IBM i 7.3, PASE stack frames will now be collected and reported in Job Watcher when the data collection option to include \*CALLSTACK data is specified.

### **2.6.5 Collecting and Analyzing the Data**

Collection Services and Job Watcher are two performance data collection mechanisms on IBM i that collect the wait accounting information. Job Watcher also collects holder and waiter information, as well as call stacks. Once the performance data is collected, you can graphically analyze the data. The iDoctor product has a Windows® client for graphically viewing performance data. IBM Navigator for i web console has Performance Data Investigator (PDI) feature to graphically view performance data through a web browser interface.

## **2.7 Acceptable Performance**

It is best to evaluate performance of a system through objective measurements, for example through application log files, or batch job run times. Acceptable performance is based on customer expectations. Expectations can be based on benchmarks, modeling, or experience. But if the actual users of the system feel that the response time or throughput is unacceptable, this is a key indication of a problem. Incorrect assumptions when architecting a system may create a situation where acceptable performance cannot be attained.

## **2.8 Partition Scaling Considerations**

IBM Power hardware is extremely versatile and scalable. One can quite easily add additional hardware resources (CPU, memory, disk, adapters, etc.) to an IBM i partition to scale the partition hardware. In many cases, the software application throughput scales linearly with the hardware. In other words, increasing or decreasing the hardware resources by some percentage results in an equivalent corresponding increase or decrease in throughput. But, of course, not all environments are created equally and the software stack, allocated hardware resources, or both may have inherent performance limitations that prevent the application from scaling linearly, up or down. This means that software application response time, throughput, or resource consumption may not scale linearly with the hardware. This can show up in any number of ways, such as requiring more of a specific hardware resource when scaling up, to requiring more than expected hardware resources when sizing a partition based on a CPW rating. There are many scenarios. And, in fact, there are scenarios where throughput could significantly drop, yet there could be an exponential demand for specific hardware resources.

These considerations are not specific to IBM i or Power hardware. While the terminology is different, here's an AIX document that discusses considerations from their viewpoint:

[Considerations in software design for multi-core multiprocessor architectures](#)

When investigating application scalability, understand that all work requested of the hardware starts with the application software. Improving the software application (e.g. eliminating software bottlenecks or inefficient coding practices) or altering the workload can typically be done more quickly than waiting for future hardware enhancements and operating system releases. The best approach to ensure that

the application software scales linearly includes the following:

- Test the environment. The test environment should consist of the same hardware resources with the same application code and operating system release running the same expected volume of work. As workload is added or if new applications are deployed, creating a test environment or benchmark is the best way to understand if an application can scale. IBM has [benchmark centers](#) to assist with this.
- Engage IBM Technology Expert Labs. [Chapter 8](#) has more information on how to engage IBM Technology Expert Labs.
- Review resource usage. Use the tools in [chapter 7](#) to understand how system resources are utilized.
- Plan for performance as workload is added. Use the tools in [chapter 6](#) to watch performance trends. Use the tools in [chapter 4](#) to plan for adding additional hardware resources as workload increases.
- Avoid common pitfalls. [Chapter 8](#) (and the following chapters) has more information on good practices to follow: reduce full opens, program activations, high logical I/O, etc.

Examples considerations include:

- Memory - Each job has a working set. That working set grows from time period to time period. The application's working set size could increase at a rate not consistent with the workload and require more memory than expected.
- Disk IO – A high volume of disk operations may eventually exceed hardware replication (e.g. global mirroring) capability, and this can cause response times to greatly increase.
- CPU - A move from slower to faster processors could result in more jobs concurrently executing a single shared data object (e.g., counter) and result in an unexpected latency and resource consumption.
- CPW rating - An equivalent sized partition on a different hardware architecture platform may not perform as expected for a given application solution.

## 3 Performance Benchmarks

### 3.1 Introduction

This chapter covers the performance benchmarks and how they should be used to compare system performance.

### 3.2 What Are Performance Benchmarks?

Performance benchmarks are well-defined tests that serve as a basis to evaluate and compare the performance of computer systems. Performance benchmark tests use representative sets of programs and data designed to evaluate the performance of computer hardware and software in a given configuration.

### 3.3 How are IBM i systems rated for performance?

IBM publishes Commercial Processing Workload (CPW) ratings for all IBM i systems to help customers compare the relative throughput capacity of systems.

### 3.4 What is the CPW rating of a system?

The CPW rating of a system is generated by using measurements of a specific workload that is maintained internally within the IBM i Systems Performance group. The CPW rating is designed to evaluate a computer system and associated software in the commercial environment. It is rigidly defined as a relative capacity metric for rough model comparisons and relative CPU consumption. It is NOT representative of any specific environment, but it is generally applicable to the commercial computing environment.

What the CPW rating is:

- Test of a range of database applications, including various complexity updates and various complexity queries with commitment control and journaling
- Test of concurrent data access by users running a single group of programs.
- Reasonable approximation of a steady-state, database oriented commercial application's relative performance.

What the CPW rating is not:

- An indication of the performance capabilities of a system for any specific customer situation
- A test of "ad hoc" (query) database performance
- A test of single-threaded (batch) application throughput (e.g. batch processing steps per minute)
- A test of single-threaded (batch) application run time or "batch window" (e.g. job completes in 4-hour batch window)

When to use the CPW rating results:

- Approximate product positioning between different systems running IBM i where the primary application is expected to be oriented to traditional commercial business uses (order entry, payroll, billing, etc.).
- The CPW metric should be used in terms of the system capacity potential (throughput) and not in terms of response time of a transaction, batch job, or query.

### 3.4.1 CPW Rating vs. Public Benchmarks

Specific choices were made in creating the CPW rating to try to best represent the relative positioning of IBM i systems. Some of the differences between the CPW rating and public benchmarks are:

- The code base for public benchmarks is constantly changing to try to obtain the best possible results, while an attempt is made to keep the base for the CPW rating as constant as possible to better represent relative improvements from release to release and system to system.
- Public benchmarks typically do not require full security, but since IBM customers tend to run on secure systems, Security Level 50 is specified for the CPW rating.
- Public benchmarks are optimized to obtain the best possible results for that specific benchmark, whereas for the CPW rating we tend to use more of the system defaults to better represent the way the system is shipped to our customers.
- Public benchmarks can use different applications for different-sized systems and take advantage of all of the resources available on a particular system, while the CPW rating has been designed to run as the same application at all levels with approximately the same disk and memory resources per simulated user on all systems
- Public benchmarks require extensive, sophisticated driver and middle tier configurations. In order to simplify the environment and add a small computational component into the workload, all the required components to drive the CPW rating have been included as a part of the overall workload.

The net result is that the CPW rating is an application model that IBM believes provides an excellent indicator of multi-user transaction processing performance capacity when comparing between members of the IBM i system families. As indicated above, the CPW rating is not intended to be a guarantee of performance, but can be viewed as a good indicator for multi-user transaction processing workloads

## 3.5 Comparing Benchmark Results

When comparing performance benchmark results, it is important to compare “apples-to-apples” by comparing the results of the same performance benchmark test(s). The result of one benchmark test often does not represent the performance of a computer system for another workload.

For example, the result of a floating point intensive benchmark test doesn't provide any information about the performance of the same computer running an integer intensive benchmark or an OLTP workload and vice versa.

A common pitfall in setting the wrong performance expectations is to look at the results of one performance benchmark and apply it to another workload. For example, if the performance of an OLTP workload improved by 50% when upgrading from machine A to machine B, it would be invalid to assume that a Compute Intensive workload would see the same improvement. Likewise, it would be invalid to assume that a single threaded workload improvement would necessarily translate into the same OLTP workload improvement.

## 3.6 Custom Benchmarks and Proof of Concepts (PoC)

Custom benchmarks are often used when a customer needs to confirm the sizing requirements of a specific application on specific hardware configurations. This may be the case when sizing tools do not provide the level of confidence needed to make purchase decisions. Many times this involves testing application scalability on multiple hardware configurations.

Custom benchmarks can range in complexity. They can be as simple as a test of end-of-day batch processing, or as complex as simulating thousands of online users accessing a multi-tiered web

application.

When a custom benchmark is performed to measure the performance of a computer system for a customer production workload, it is important that the benchmark test represents the real workload to get meaningful data.

For example, running a single database job at a time on an otherwise idle database server provides good information about the performance of the database server under best possible conditions. However, it does not provide any information about the performance of this database job when the server is under medium or heavy production workload.

### **3.6.1 Considerations**

When doing a custom benchmark or Proof of Concept (PoC) it is important that the test is constructed to simulate the production environment. This is especially true as the hardware continues to evolve into the multi-core era and more time is being invested in the cache/memory hierarchy.

### **3.6.2 Common Pitfalls for Custom Benchmarks**

The most common pitfall when running a custom benchmark or a proof of concept is that the benchmark test does not simulate the real production environment and the benchmark result does not represent the performance the system will achieve in the production environment. The achieved benchmark result might be much better for the benchmark test than for the real production workload, which most likely would lead to performance problems later on when running the real workload. It also could be the other way around potentially causing delays or failure of the PoC.

### **3.6.3 IBM i Performance & Scalability Services Center**

The IBM i Performance & Scalability Services Center in Rochester, MN can provide facilities, hardware, and technical expertise to assist you with a custom benchmark or Proof of Concept.

Following are some examples of the types of testing that is done at the Performance & Scalability Services Center:

- Proofs of Concept (e.g. HA alternatives, SSD analysis, external storage, etc.)
- Stress test-specific hardware configuration
- Evaluate application scalability
- Performance optimization and tuning
- Assess application performance when migrating to a new release of IBM i
- Determine impact of application changes
- Virtualization, consolidation, migration services
- Capacity planning

To request a custom benchmark or Proof of Concept, submit a request by using the following link:

[Contact IBM Technology Expert Labs](#)

## 4 Sizing a system

### 4.1 Introduction

Sizing a system, and all its various components, so that it is capable of adequately supporting a production environment can be quite complex. It requires a good knowledge of the characteristics of the workload(s) to be run, and the load that they place on the system components.

Some questions to consider before beginning the sizing exercise:

1. What are the primary metrics, e.g., throughput, latency, that is used to validate that the system is meeting performance requirements?
2. Does the workload run at a fairly steady state, or is it bursty, thereby causing spikes in load on certain system components? Are there specific criteria, e.g., maximum response time that must be met during the peak loads?
3. What are the average and maximum loads that need to be supported on the various system components, e.g., CPU, memory, network, storage?

### 4.2 Sizing Best Practices

- Know your bottlenecks. If your workload is I/O constrained, upgrading to the latest processor technology will not help much. Tools such as Performance Data Investigator (PDI) and IBM iDoctor for IBM i (iDoctor) can be used to identify partition level and job level bottlenecks.
- Size for peaks to ensure you have enough capacity to have good performance when you reach end of month or year-end spikes of activity.
- Focus on your most important jobs or subsystems, not just on the overall activity on the partition. While adding cores will reduce CPU utilization of the partition, it may not improve the performance of your batch jobs for example.
- Be careful with I/O sizing tools. IBM i application performance is sensitive to disk write response time and the number of disk units or LUNs. Moving from internal to certain external storage solutions, and/or greatly reducing the number of disk units/LUNs may negatively affect your application's performance.

### 4.3 System Sizing Tools

#### 4.3.1 IBM Systems Workload Estimator (WLE)

The IBM Systems Workload Estimator (WLE) is a web-based sizing tool for IBM Power and IBM z Systems. WLE is available at:

<https://wle.c8f8f055.public.multi-containers.ibm.com/wle/EstimatorServlet>.

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. WLE characterizes your projected workload either with customer measurement data or by using one of the many workload plug-ins (a.k.a., sizing guides). Virtualization can be reflected in the sizing to yield a more robust solution by using various types of partitioning and virtual I/O. WLE provides current and growth recommendations for processor, memory, and disk (either internal or SAN) that satisfy the overall customer performance requirements.

WLE can support sizings dealing with multiple systems, multiple partitions, multiple operating systems, and multiple sizing intervals. These features can be coordinated by using the functions on the Workload Selection screen.

WLE recommends the system model, processor, memory, and disk requirements that are necessary to handle the overall workload with reasonable performance expectations. To use WLE, you select one or more workloads and answer a few questions about each workload. Based on the answers, WLE generates a recommendation and shows the predicted CPU utilization of the recommended system in graphical format. The results can be viewed and saved as a PDF. The visualize solution function can be used to better understand the recommendation in terms of time intervals and virtualization. For many systems, WLE can also provide an energy consumption estimate to help with your "go green" initiatives. There is an optional integration point with planning tools (System Planning Tool, and the IBM Pre-sales Advisor Tool) so that the configuration planning and validation may continue.

The WLE recommendation is based on processing capacity, which assumes that the system can handle the aggregate transaction rate and that the application can fully scale on the system. Although WLE does not model response times specifically, it abides by the best practice utilization guidelines to help minimize potential negative impacts to response time. Beyond what is recommended here, additional system resources may be required for additional workloads not sized here, growth resulting from workload changes, version/release changes not already considered, minimum memory to support I/O or virtualization configurations, minimum disk to support multiple ASP or RAID configurations, and all other resources beyond the scope of WLE (CPU, memory, disk).

The WLE recommendation assumes that your system is well tuned in terms of performance (including the system hardware configuration, operating system settings, virtualization configuration and settings, and the application). The WLE scaling algorithms assume that the sized applications are multi-thread capable and are able to exploit and scale with multiple cores and SMT; otherwise, the sizing is invalid. So, do not use WLE to size single-threaded applications or for single-threaded time critical batch jobs. For these, it is important to also consider the performance per thread and per core, as well as GHz.

Sizing tool input starts with well-defined, realistic, consistent workloads and not with industry benchmarks (that many times avoid customer-like functions like logging and random non-cached accesses). WLE workload requirements come from performance data from existing customer systems or from WLE workload plug-ins. In the case of existing systems, it is important to consider peak data to fully support your business along with a growth trend. For plug-ins, it is important to answer the questionnaires with responses to adequately cover your peak requirements.

As with every performance estimate (whether a rule of thumb or a sophisticated model), you always need to treat it as an estimate. This is particularly true with robust IBM Systems that offer so many different capabilities where each installation will have unique performance characteristics and demands. The typical disclaimers that go with any performance estimate ("your experience might vary...") are especially true. We provide these sizing estimates as general guidelines but can't guarantee their accuracy in all circumstances.



#### **4.3.1.1 How can I get the IBM Workload Estimator?**

IBM Workload Estimator can be found at: [Estimator](#)

#### **4.3.1.2 How can I get assistance in sizing a system?**

Utilize the IBM Workload Estimator or contact your IBM Business Partner or IBM Sales. IBM Post Sales Support is not able to assist with sizing questions.

#### **4.3.2 *IBM Storage Modeller***

Storage Modeller is a cloud-based storage sizing tool accessed through a web browser that models IBM SAN storage solutions for customer applications. This tool is designed for use by technical sellers of IBM storage systems, both IBM Sales and IBM Business Partners.

For more information about the tool, visit: [IBM Storage Modeller](#)

To access the tool, visit: <https://www.ibm.com/tools/storage-modeller>

## 5 Proactive Performance Monitoring

### 5.1 Introduction

This chapter covers proactive performance monitoring process from a high-level point of view. Its purpose is to provide a guideline and best practice on how to address performance problems by using a top down approach.

### 5.2 Performance monitoring before a problem occurs

Application performance should be recorded using log files, batch run times, or other objective measurements. General system performance should be recorded and should include as many components of the environment as possible.

Performance statistics should be collected based on the typical period of activity in your environment. The minimum period should be considered one week in an interactive system, as typical performance may vary based upon the day of the week. A batch processing system that typically runs large end of month reports should consider a month as the minimum period. Data should be collected and stored over multiple periods to allow for comparison.

Performance monitoring of the environment provides many benefits. The application performance provides a useful method of quantifying the performance change. A history of acceptable system performance focuses analysis on changed components. Also, continual review of performance monitoring will show trends in performance and capacity before they become system-wide problems.

#### 5.2.1 *Real-time Interactive Monitoring*

Basic system commands provide a real-time view of data and can help you identify potential problems early. The most common are:

**WRKACTJOB** -- shows all jobs with key system resources used (including CPU %, response time, I/O information, and temporary storage used by each job.)

**WRKSYSSTS** -- shows system utilization and pool usage

**WRKDSKSTS** -- shows statistics for all disk units

**WRKSYSACT** -- shows tasks, primary threads and secondary threads that have consumed CPU in the refresh period. Also shows temporary storage used.

Except for WRKSYSACT, equivalent GUI task versions of these commands are available in IBM Navigator for i (Active Jobs, System Status, and Disk Status)

In addition, corresponding IBM i Services are available ([ACTIVE\\_JOB\\_INFO](#), [SYSTEM\\_STATUS](#), and [SYSDISKSTAT](#)).

#### 5.2.2 *Dashboard on Navigator for i*

The dashboard on [IBM Navigator for i](#) provides real-time information for multiple systems.

## IBM Power Performance

Data is updated automatically by default. Each system added will display the same three metrics: CPU percent used, System ASP percent used, and number of Active Jobs.

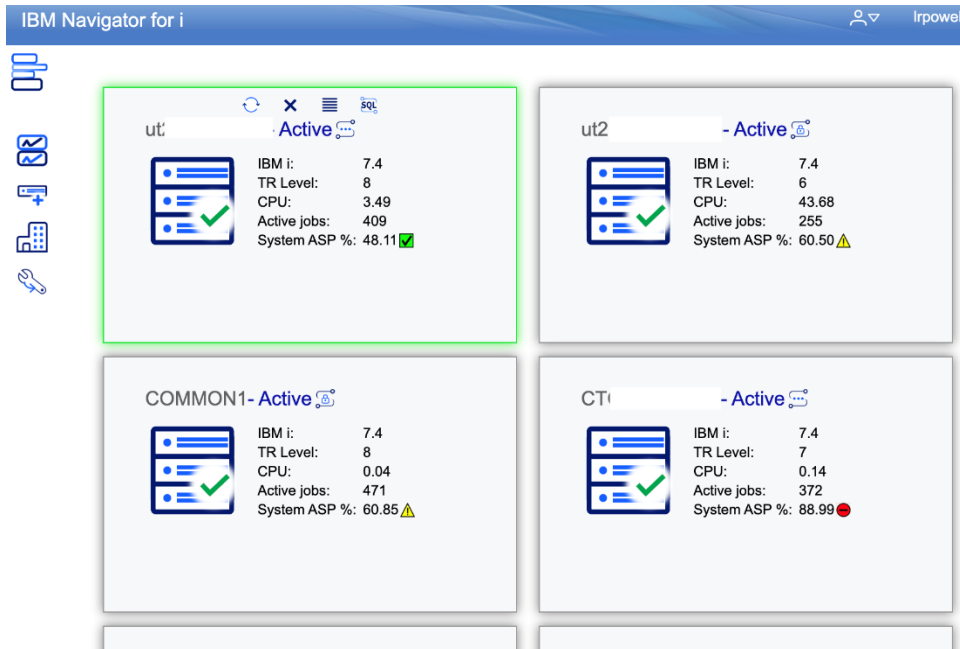


Figure 1 - IBM Navigator for i Dashboard

Add more nodes to see more systems on the managing node GUI dashboard. Select the upper-left icon to switch from tile view to table view.

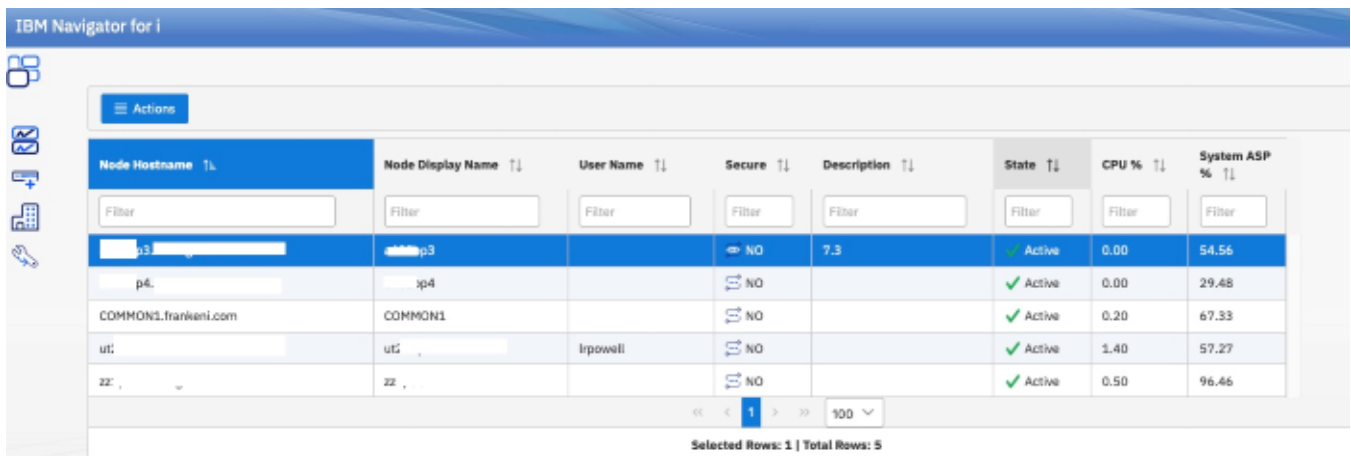


Figure 2 - IBM Navigator for i Dashboard Table View

For additional details on this dashboard, see Navigator for i - [Dashboard](#)

### 5.2.3 Automated Monitoring

There are monitoring services available in **IBM Navigator for i**.

#### 5.2.3.1 IBM Navigator for i

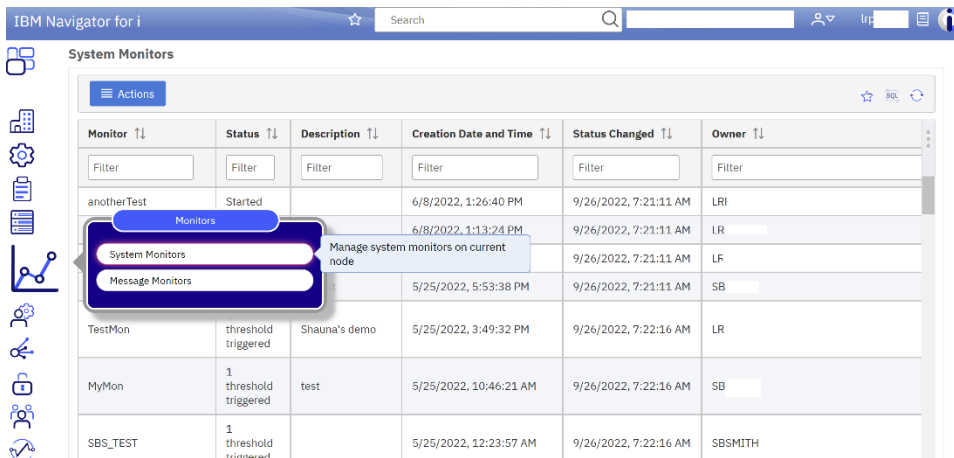
[IBM Navigator for i](#) is a modern web-based interface for managing and monitoring one **or more** IBM i instances from a single location.

[System monitors](#) and [Message monitors](#) are available in IBM Navigator for i.

[Monitors](#) can be used to pro-actively monitor system health and message queues, alerting you of potential performance problems before they become serious issues.

Thresholds are set for various metrics on these monitors and provide a command action (optional) which is executed when a monitor detects that a threshold is triggered. For a command example, you can run an IBM i command or start a program when the threshold is triggered.

System monitors have a collection interval that is set from 15-300 seconds (5 minutes). Generally, for system monitors, 60 seconds provides a good balance between timeliness and the size of the monitor data collection. 15-second collections can get large quickly.



**Figure 3 - IBM Navigator for i System Monitors**

The system monitors in IBM Navigator for i use Performance Data Investigator to display the monitor performance data as a graph or multiple graphs on a dashboard view. These system monitors are limited to a single partition.

IBM Navigator for i system monitor metrics are collected, calculated, and managed by Collection Services and the data is stored in database files (no longer stored in a private \*MGTCOL as was done in Management Central).

Because Collection Services calculates and manages the system monitor metric data, the metric data can be generated for your Collection Services collection without ever starting an IBM Navigator for i System Monitor. To do this, specify CRTPFERSUM(\*ALL) on either the CFGPFCOL or CRTPFRTA

command. Or you can start a system monitor collection by using the CFGPFCOL command and specifying ENBSYMON(\*YES).

For more information, see:

[IBM Docs - IBM Navigator for i Monitors](#)

[System Monitors Overview](#)

[Message Monitor](#)

### 5.2.4 *Watching for messages*

The system provides a watch-for-event function that allows you to watch for messages. The watch-for-event function allows you to specify messages for which you want the system to watch. When these messages occur, a user exit program is called to take any necessary action. You must specify the message queue or job log where you expect the message to be sent. You can specify some text string to be compared against the message data, the from-program, or the to-program. You can also select watches by message type or severity. Using this watch selection capability allows you to be informed of only certain messages. Watch sessions can be started, ended, and displayed using IBM Navigator for i. For more information, see:

[Watching for messages](#)

### 5.2.5 *IBM i Services*

Many services are available that can help you monitor key aspects of your partition. IBM Services can be listed and launched to from IBM Navigator for i. For more information, see:

Chapter 9: [IBM i Services](#)

[IBM i Services \(SQL\)](#)

## 5.3 Defining a Performance Problem (What is slow?)

Before collecting any additional data or making tuning or configuration changes, define what exactly is slow. A clear definition about what aspect is slow usually helps to shorten the amount of time it takes to resolve a performance problem and gives a performance analyst a better understanding of what data to collect and what to look for in the data.

A good way to define a performance problem is to answer questions like:

- Is everything slow or just a specific task?
- What application log file demonstrates the performance problem?
- Is it only slow when initiated remotely but runs fast locally?
- Is it slow for everyone or just for a single user?
- Etc.

## 6 Performance Data Collectors and Analysis Tools

### 6.1 Introduction

This chapter is intended to provide information on which tools are available to **collect** data on IBM i systems and the toolsets available to **visualize** the data, so you can more easily interpret and analyze the data.

### 6.2 Managing performance on IBM i

Managing performance requires the use of a variety of specialized applications. Each of these applications offers a specific insight into system performance. Some applications collect the data, while others are used to display, analyze, monitor, or manage the data collected.

For more information, see the following link: [IBM Docs: IBM i Performance](#)

### 6.3 Performance Data Collectors

In the area of performance management, a key strength of IBM i is the collection technologies built into the operating system. The rich metric instrumentation provided can then be used by toolsets to aid in data analysis.

The primary data collectors include:

- Collection Services
- Disk Watcher
- Job Watcher
- Performance Explorer
- SQL Plan Cache

Each collector has a unique value-add and properties, including the type of data it collects - typically sample data or trace data. Sample data refers to metrics that are collected at some time interval. Trace data refers to data that is captured as system or application trace points are hit. Trace data records everything that happened, while sample data provides a summary of what took place during each sample interval.

For information on performance data collection tools, see the [Performance Data Collectors](#) section in the IBM Docs.

Recommendations for configuring the performance data collectors are included in the sections below.

#### 6.3.1 *Collection Services*

Collection Services provides for the collection of system and job level performance data. It is the primary collector of system data and IBM recommends you always run with Collection Services active.

Collection Services collects data from many system resources such as:

- CPU
- Memory pools
- Disk (internal and external)

- Communications

Collection Services samples system and job level performance data with low overhead. It can collect data at regular intervals from 15 seconds to 1 hour (default is 15 minutes).

Often, Collection Services is the first analysis tool used to:

- Monitor how your system is running
- Investigate a reported problem
- Look for improvement candidates

Use Collection Services to understand resource usage, what is or is not being used, how did it change, when did it happen and what is affected.

Best practices for Collection Services:

- Collect with 5-minute intervals
- Keep at least one week of data
- Keep historical collections for comparison
- Enable cross partition performance collection on the HMC (see [Collecting and Viewing Performance Data for Entire Physical System](#))

Collection Services can be configured and managed through the IBM Navigator for i Performance interface, IBM iDoctor for i, or CL commands. For more information, see: [IBM Docs: Collection Services](#)

### 6.3.2 *IBM i Disk Watcher*

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

Disk Watcher obtains data on I/O operations to disk units, along with 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.

The Disk Watcher performance data collector was added to the IBM i operating system with the 6.1 release.

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 respreading data across units
- Determining the optimal placement of devices, IOAs, or buses

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

For more information, see the following:

- [IBM Docs: IBM i Disk Watcher](#)

### 6.3.3 *IBM i Job Watcher*

IBM i Job Watcher provides for the collection of job data for any or all jobs, threads, and tasks on the system. It provides call stacks, SQL statements, objects being waited on, Java™ 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. The Job Watcher performance data collector was added to the IBM i operating system with the 6.1 release.

Best practices for Job Watcher:

- Consider running Job Watcher continuously
  - [QMGTOOLS: Job Watcher Monitor Function](#)
  - [iDoctor Monitors Guide](#)
- Use a definition with 5 or 10 second intervals, collect SQL information, and collect call stacks (Q10SECSQL for example)
- Keep historical collections for comparison

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

For more information, see the following: [IBM Docs: IBM i Job Watcher](#)

### 6.3.4 *Performance Explorer (PEX)*

Performance Explorer collects 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 by using Collection Services, Job Watcher, Disk Watcher 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

As your computer environment grows both in size and in complexity, it is reasonable for your performance analysis to gain in complexity as well. Performance Explorer addresses this growth in complexity by gathering data on complex performance problems.

Note: Performance Explorer is the tool you need to use after trying the other tools. It gathers specific forms of data that can more easily isolate the factors involved in a performance problem. However, you can significantly affect the performance of your system when you collect this data – especially if you collect this data improperly.



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.

Performance Explorer can be configured and managed through IBM iDoctor for IBM i or CL commands.

For more information, see the following: [IBM Docs: IBM i Performance Explorer](#)

In 2007, a Redbook was written about using PEX for analyzing application performance. Even though the content was written for the V5R3 and V5R4 releases, much of the content is still relevant (except for screen captures and iDoctor graphical user interface experience). The Redbook can be found here :

[Application and Program Performance Analysis Using PEX Statistics](#)

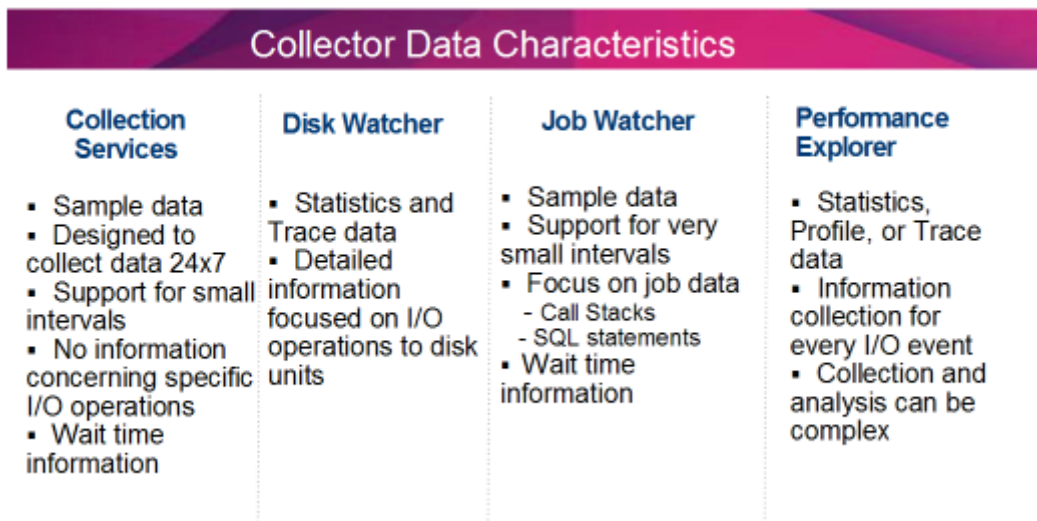


Figure 4 - Collector data characteristics

### 6.3.5 SQL Plan Cache

The plan cache contains access plans for SQL queries that have already run or are currently running on the partition. It also includes performance information such as query run times, CPU usage, and I/O usage. The plan cache does not survive an IPL, so it is a best practice to generate a plan cache snapshot before each IPL and before any major application or software changes. A snapshot can be generated by using either the stored procedure CALL QSYS2.DUMP\_PLAN\_CACHE or through IBM i Access Client Solutions (ACS) SQL Performance Center.

## 6.4 Analysis Tools

### 6.4.1 Overview

It's important to understand the difference between the **performance data collectors** and the **visualization** or analysis of that data. The performance data collectors are part of the operating system, and anyone can collect all types of performance data. However, to visualize or analyze that data may

require additional interfaces.

IBM provides several tools with various user interfaces for viewing, analyzing, reporting, graphing, and managing performance data.

### 6.4.2 *Performance Tools for i*

The [IBM Performance Tools \(PT1\) LPP](#) contains many tools including performance reports for Collection Services. The performance reports are text-based and may not be the easiest way to analyze performance data; graphical analysis is often easier.

The requirement to install PT1 to access certain features of Performance Data Investigator (PDI) graphical analysis has been removed. Any authorized user can now access and use the following features: Disk Watcher, Performance Explorer (PEX), Database, and Job Watcher.

The IBM Performance Tools (PT1) LPP is now available at no charge. Refer to [IBM i Portfolio Simplification](#) for additional details.

### 6.4.3 *IBM i Access Client Solutions (ACS)*

ACS provides a Java based, platform-independent interface that runs on most operating systems that support Java, including Linux, Mac, and Windows™. It consolidates the most used tasks for managing your IBM i into one simplified location. It includes many important database functions such as SQL Performance Center and Run SQL Scripts. It also includes the ability to launch Navigator for i.

### 6.4.4 *IBM Navigator for i*

The Navigator is the strategic application for IBM i administration tasks. Navigator supports most tasks that are needed to manage and monitor one or more IBM i instances from a single location.

It includes functions such as Performance Data Investigator (PDI), which is a powerful tool used to analyze performance data (described further in section 6.4.5). Nothing needs to be installed on your workstation to use IBM Navigator for i. The Web application is part of the base IBM i operating system and can be easily accessed by pointing your browser to: <http://hostname:2002/Navigator/login> where "hostname" is the name of your server and "2002" is the port number. This is an insecure connection. For a secure (TLS) connection (recommended) use <http://hostname:2003/Navigator/login>.

It is available on IBM i 7.3, IBM i 7.4, and IBM i 7.5. The IBM i 7.3 and IBM i 7.4 [PTFs](#) were first released in September 2021.

For more information refer to:

[Navigator for i](#)

[IBM Docs: IBM Navigator for i](#)

[IBM Support Documentation: IBM Navigator for i - TLS Encryption](#)

### 6.4.5 Performance Data Investigator

Performance Data Investigator (“PDI”, a function included in IBM Navigator for i) provides a web-based GUI for performance data with interactive charts and tables. With PDI you can view and analyze performance data for each of the collectors (Collection Services, IBM i Job Watcher, and IBM i Disk Watcher). PDI also contains charts for Health Indicators, Database, System Monitors, and Graph History (available in IBM i 7.3 and later releases).

Other useful performance tasks found under Performance include configuration commands for Collection Services, Disk Watcher and Job Watcher, and Managing Collections.

PDI is accessed by selecting the Performance category, then opening the **Investigate Data** task. From here, you can access the many perspectives (charts and tables) grouped by package name:

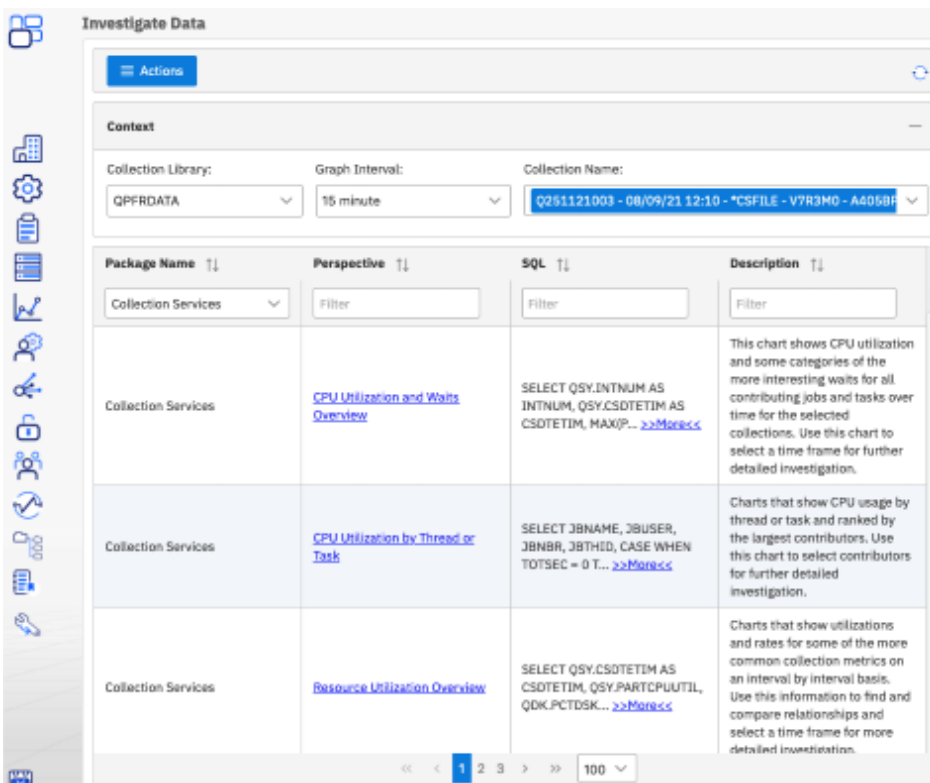


Figure 5 - Performance Data Investigator

The capability to view perspectives for all packages is included with the base operating system: Collection Services, Health Indicators, System Monitors, Job Watcher, Disk Watcher, Database, and Graph History.

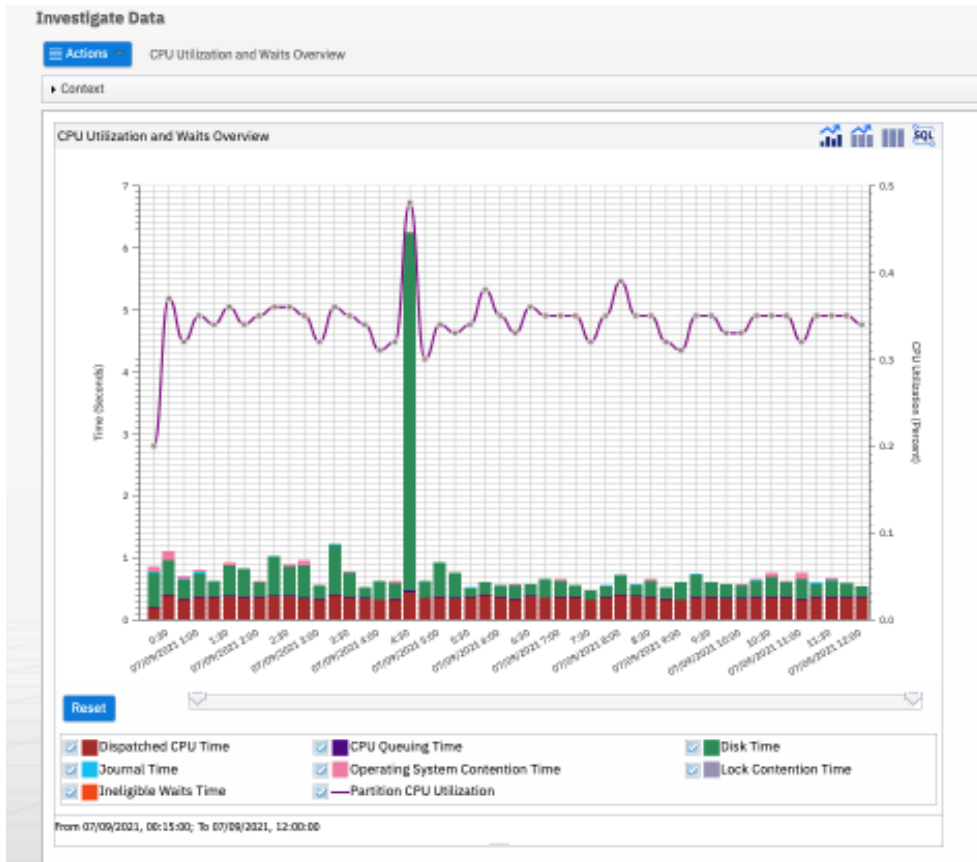


Figure 6 - Performance Data Investigator example

Graph History charts use Collection Services historical data that is available in IBM i 7.3 and later.

In the Graph History window, you can see a graphical view of the metrics collected by Collection Services for an extended period (multiple days, months, and years). You can contrast this data with the real-time data shown in a System Monitor window.

Shown below is a stacked chart example where one week’s worth of data is stacked so that each day can be compared. Also shown on this page are two drilldown panels on the right side. The top-right panel is the drilldown information with the top contributors for one data point selected on the stacked chart. The bottom-right panel is the properties information for one object selected from the top contributors list.

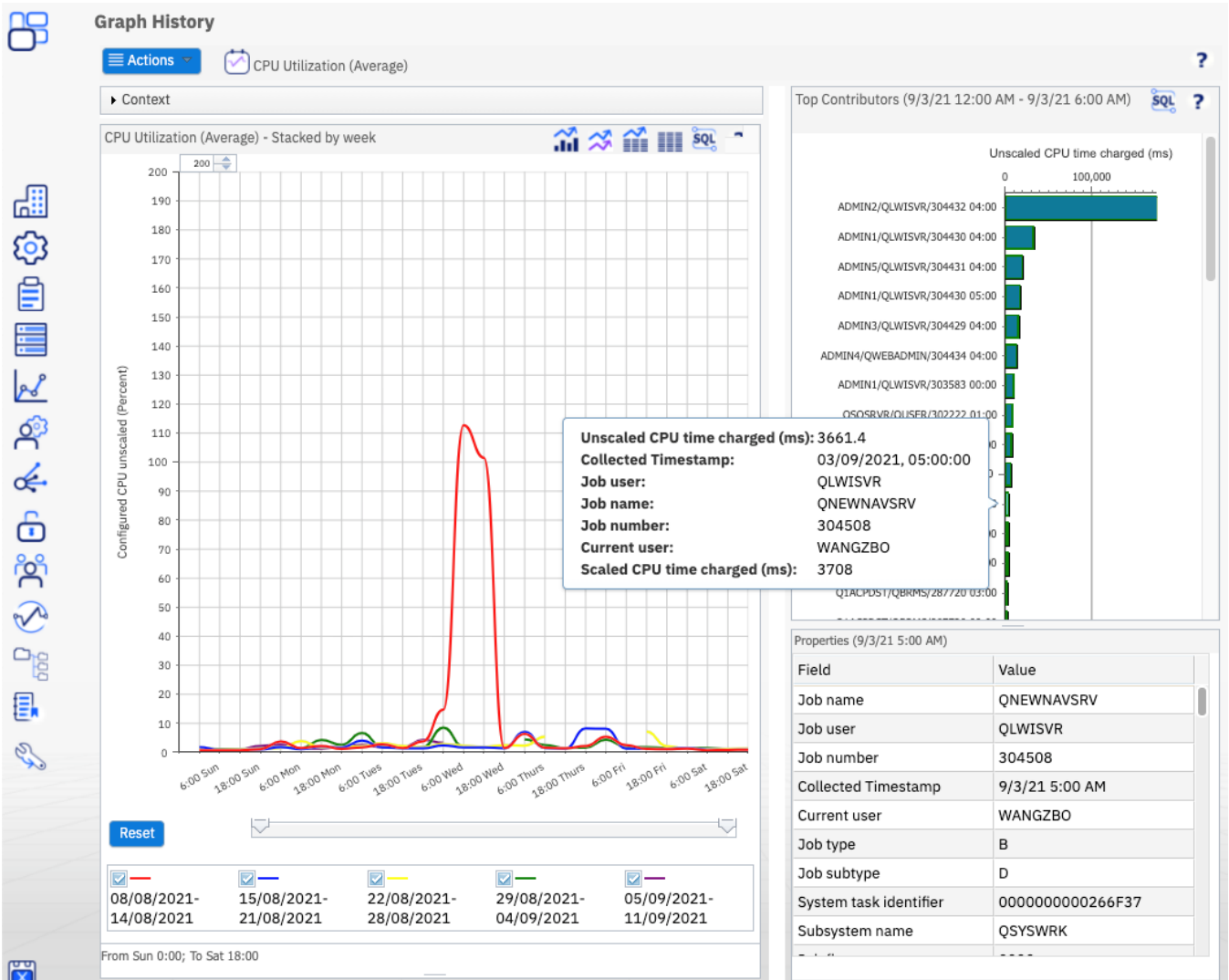


Figure 7 - Graph History example

For more information see the following links:

- [A new way to analyze historical performance data on IBM i](#)
- [IBM Docs: Performance Data Investigator](#)
- [Performance on the web - Performance tools for IBM i](#)

### 6.4.6 IBM iDoctor for IBM i

IBM iDoctor for IBM i is a Windows-based suite of performance tools that can be used for analysis of Collection Services, Job Watcher, PEX, Disk Watcher, and SQL Plan Cache Snapshots.

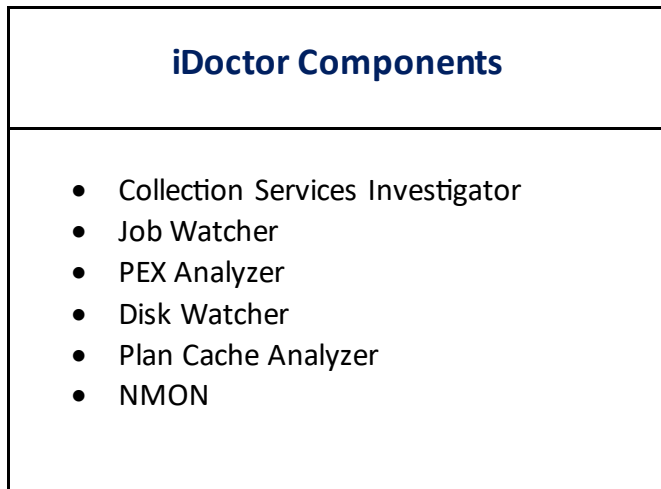


Figure 8 - iDoctor components

IBM iDoctor for IBM i is generally used by the performance expert to collect, investigate, and analyze performance data on IBM i. The tools are used to monitor overall system health at a high "overview" level or to drill down to the performance details within job(s), disk unit(s) and/or programs over data collected during performance situations.

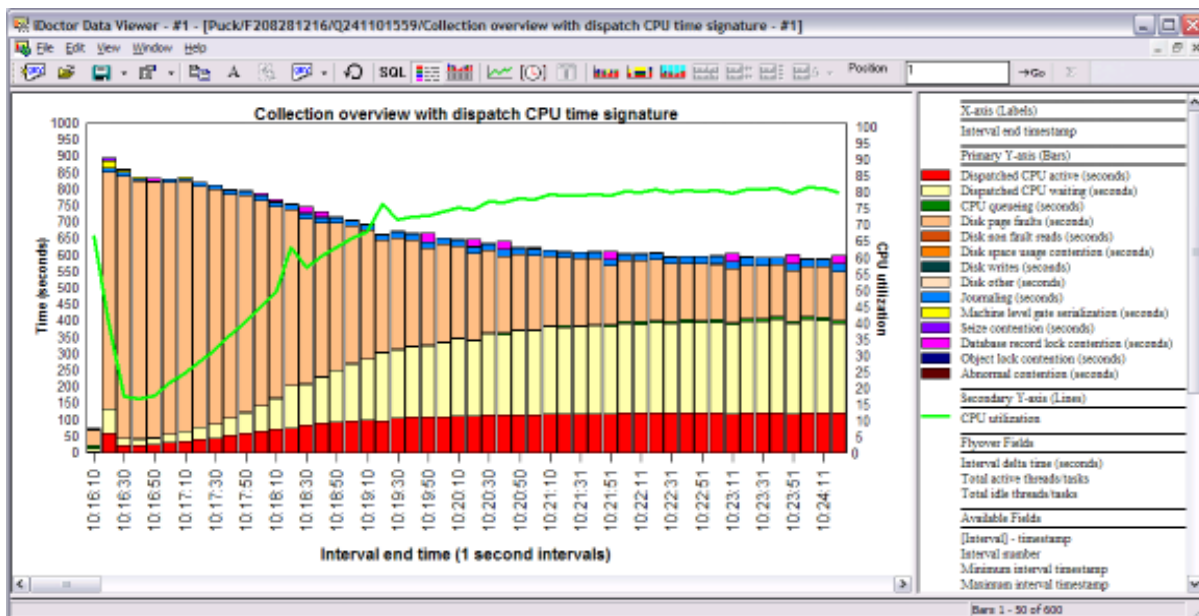


Figure 9 - iDoctor example

iDoctor is constantly evolving to meet the needs of our clients. Therefore, it is not a formal licensed program product. The software is licensed for use under a service agreement. For more information, a free 45-day trial license, or a price quote, visit the following link: [IBM iDoctor for IBM i website](#)

Training for iDoctor is available. For more information, refer to this link: [iDoctor Workshop](#). For

additional information on Education and Training, refer to section of chapter 16.

**6.4.7 Tools Matrix**

The following chart shows at a high-level the data that can be analyzed by the various IBM i performance analysis tools.

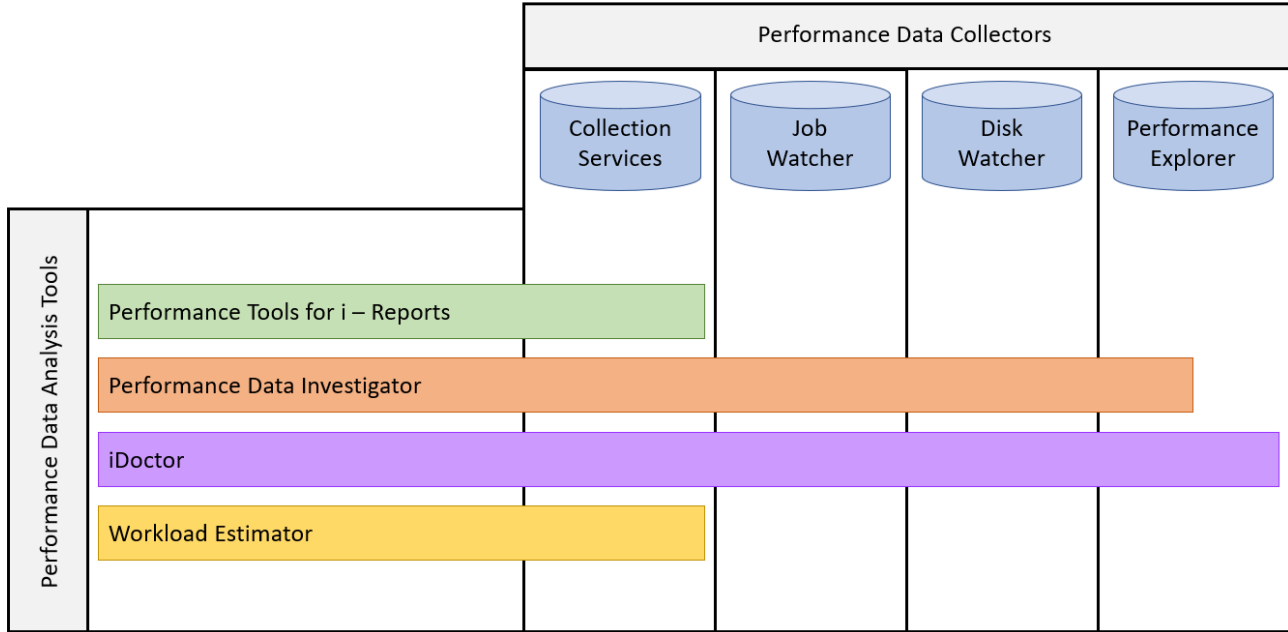


Figure 10 - Relationship between collector and major functions / interfaces

**6.4.8 Open Source Languages**

Generally speaking, all of the analysis tools discussed in this section can be used to investigate programs written in open source languages, including those running in PASE. Examples of such languages include JavaScript (Node.js), Python, Ruby, PHP, and others. Performance investigation of these programs can also leverage language-specific tools. The open source community often maintains tools that can perform method profiling, memory analysis, tracing, etc., and these tools can be used complementary to IBM i tooling.

## 7 Frequently Asked Questions

### 7.1 Introduction

This chapter covers frequently asked questions.

### 7.2 General Questions

#### 7.2.1 *What are the general performance tuning recommendations for best IBM i performance?*

There is no such thing as a general performance tuning recommendation.

#### 7.2.2 *I heard that... something on the internet said... should I change...?*

No, never apply any tuning changes based on information from unofficial channels and always consider the author or source of that information. Make sure the advice applies to IBM i, especially when related to Java and Db2. Configuration and application changes should be done based on performance analysis and tested before being made in production.

#### 7.2.3 *My <CPU, disk, memory> usage is high, should I add more resource?*

Before adding more hardware, ensure the resources you have are being utilized efficiently. System tuning, database, and application analysis often can result in significant improvements in performance.

#### 7.2.4 *Why didn't the performance of my partition improve after I added <CPU, memory, disk arms, SSDs>?*

The bottleneck may not be related to the resource you added (CPU, disk, memory). Proper wait time analysis should be done to identify the bottleneck and how to best resolve it. See the [Components of Response Time](#) section earlier in this document.

#### 7.2.5 *Should I update PTFs before undertaking a performance analysis project?*

Yes! There are often PTFs that resolve performance problems you may be experiencing. You should install the latest cumulative and group PTFs and technology refresh (TR) levels.

[IBM i Technology Refreshes](#)

and

[IBM i Support: Recommended fixes](#)

Some PTFs are required before certain performance data can be collected. In addition to the latest performance tools group PTFs (SF99723 for 7.3, SF99663 for 7.4, and SF99953 for 7.5), the following lists should be checked for additional PTFs:

#### 7.5

[Job Watcher, Performance Explorer \(PEX\), and Disk Watcher PTFs for IBM i 7.5](#)

#### 7.4

[Job Watcher, Performance Explorer \(PEX\), and Disk Watcher PTFs for IBM i 7.4](#)

#### 7.3

[Job Watcher, Performance Explorer \(PEX\), and Disk Watcher PTFs for IBM i 7.3](#)

You can find information on the latest Db2 for IBM i fixes and enhancements at the following link. IBM recommends staying current on fixpacks, which can include performance enhancements.

[Db2 for i - Technology Updates](#)



You should also keep the firmware level current as this can also affect overall system performance. Additionally, certain performance data is only collected if you are at a minimum firmware level.

Released PTFs enhance the functionality of the features in Navigator, including the performance tasks. Have the latest levels of the following groups for full functionality of the Navigator tasks:

- Performance Tools group
- HTTP Server group
- Java group
- Database group

**7.2.6 How can I find out which jobs are consuming the most temporary storage on my partition?**

Before the 7.2 release, this was a very tedious job. In 7.2, IBM made many improvements to address these difficulties. The most significant is how temporary storage is tracked. How temporary storage is allocated (to a job, independent of a job, shared between jobs) does not change, but how the system keeps track of where temporary storage is used is vastly improved. The system now has a concept of temporary storage buckets that can tell you exactly where all the temporary storage is being used. The enhancements include being able to see temporary storage used by a job in Active Jobs in IBM Navigator for i, WRKACTJOB, or by using the System Temporary Storage Service ([SYSTMPSTG](#)).

In Navigator for i **Active Jobs**, by adding in this new column and sorting in a descending fashion, you can now see the top consumers of temporary storage:

The screenshot shows the 'Active Jobs' page in IBM Navigator for i. A table lists various jobs with their details. A red arrow points to the 'Active Jobs' header, and another red arrow points to the 'Temporary Storage Used(MB)' column header. The table is sorted by this column in descending order.

Job Name	Detailed Status	Current User	Enhanced Type	CPU %	Run Priority	Thread Count	Temporary Storage Used(MB)
ADMIN2	THDW	QLWISVR	BATCH_IMMEDIATE	0.25	25	144	920
ADMIN1	THDW	QLWISVR	BATCH_IMMEDIATE	0.95	25	290	733
ADMIN5	THDW	QLWISVR	BATCH_IMMEDIATE	0.02	25	76	182
ADMIN4	THDW	QWEBADMIN	BATCH_IMMEDIATE	0	25	83	177
ADMIN3	THDW	QLWISVR	BATCH_IMMEDIATE	0	25	76	164
QUSRDIR	SIGW	QDIRSRV	BATCH	0	50	12	148
QTCPPWRK	DEQW	QSYS	SYSTEM	0	20	2	144
QINAVMNSRV	THDW	QLWISVR	BATCH_IMMEDIATE	0.03	50	52	85
QZDASOINIT	TIMW	CHANGZH	PRESTART_BATCH	0	20	1	85
QYSPFRCOL	DEQW	QSYS	BATCH	0.01	1	28	80
QZDASOINIT	PSRW	QUSER	PRESTART_BATCH	0	20	1	77
QNAVMSRV	THDW	QLWISVR	BATCH_IMMEDIATE	0.04	50	49	76
QZDASOINIT	TIMW	WANGZBO	PRESTART_BATCH	0	20	1	59

Figure 11 - Active Jobs Temporary Storage Used - Navigator for i

The System Temporary Storage Service provides a programmable interface to display which active jobs are the top consumers of temporary storage:

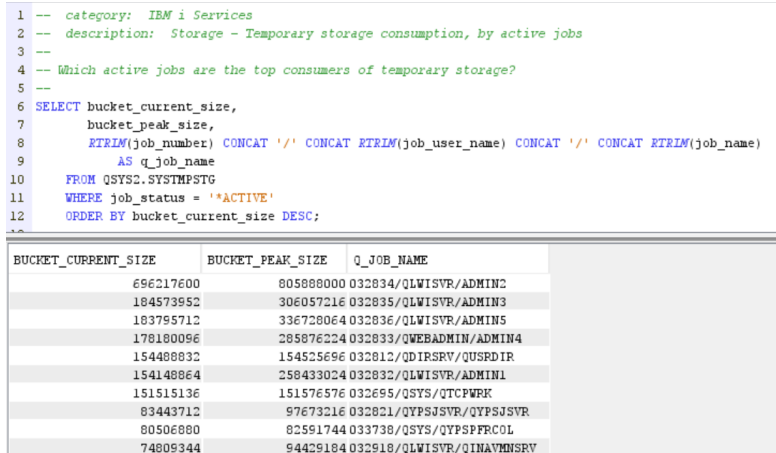


Figure 12 – System Temporary Storage Consumption by Active Jobs - IBM i Services

In addition, the system temporary storage global buckets can be viewed in Navigator for i by selecting the System icon on the left side and selecting Temporary Storage from the menu.

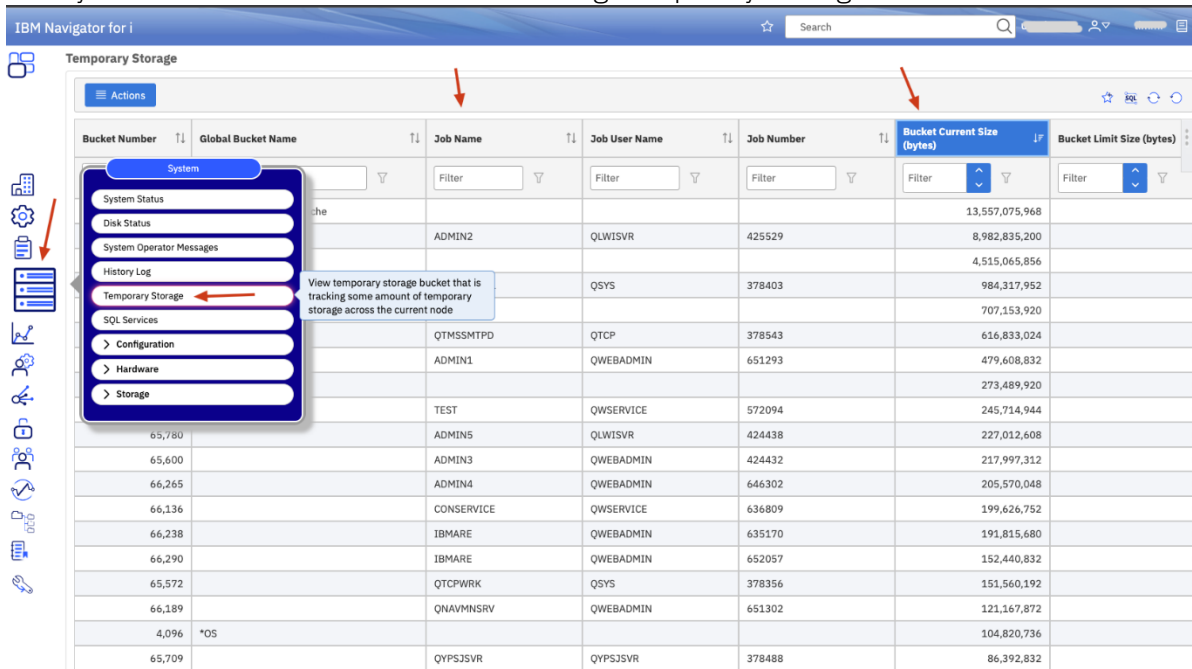


Figure 13 – Temporary Storage Used - Navigator for i

**7.2.7 There are two graphical interfaces for IBM i performance data – Performance Data Investigator (PDI) and iDoctor. Which should I use?**

Like all performance-related questions, it depends. IBM i is an industry leader with its rich integrated [Performance Data Collectors](#). To visualize all the various data collected in meaningful ways, IBM has two graphical user interfaces available, Performance Data Investigator (PDI) and iDoctor. Both GUI

tools provide visualization over the various types of IBM i performance data.

In general, PDI is geared toward the average user and iDoctor is geared to the expert user who does performance diagnostics frequently. PDI is more limited in features and functions but is easier to learn and use. PDI is included with the operating system, and you can visualize Collection Services data with no additional products or cost. iDoctor has many advanced features and capabilities and tends to be a more complex tool. However, for deep diagnostics, iDoctor can provide insights into the data where PDI is more limited.

A key advantage of PDI is that it is a task within Navigator for i. This means everyone has PDI and it has valuable integration with other IBM i management tasks. For example, you can launch PDI from active jobs, database tasks, System Monitors, and Graph History. iDoctor is a separate user interface that does not have this level of integration with Navigator management tasks.

The table below summarizes the key differences between PDI and iDoctor to help you determine which interface is best for you.

Feature	PDI	iDoctor
User interface	Browser No client OS restrictions	Windows application
IBM i Performance Data Supported	Collection Services Job Watcher Disk Watcher Graph History	Collection Services Job Watcher Disk Watcher Performance Explorer Graph History
Update Frequency	Quarterly	Quarterly
Support	Normal product support	Email <a href="mailto:idoctor@us.ibm.com">idoctor@us.ibm.com</a> for support
National Language Support	Translated	English only
License	PDI is part of the base OS	Yearly license terms and charges

Figure 14 - PDI and iDoctor feature comparison

### 7.2.8 What is the job QDBFSTCCOL running on my partition and why is it taking resources?

QDBFSTCCOL is an IBM i system job responsible for collecting statistics for the SQL Query Optimizer. It is enabled by the QDBFSTCCOL system value, with the default value being \*ALL. This job runs continuously at a low priority on your partition collecting valuable information on how best to implement your queries. In general, IBM does not recommend that you turn it off (a value of \*NONE can degrade query performance). There are times when you may notice this job taking more CPU than usual. There are several reasons for this, including:

- 1) A new file being introduced to the partition
- 2) Existing file statistics are inaccurate due to records being added or changed
- 3) Running a new set of queries on the partition

Viewing the job log for the QDBFSTCCOL job can provide clues on activity.

For additional information: [Why is the QDBFSTCCOL Taking CPU?](#)

### **7.2.9 How can I better isolate PASE jobs for analysis?**

PASE jobs are commonly launched from SSH sessions, QSHell, or Qp2Term. This can result in many jobs having the same name, for instance QP0ZSPWP. This can make it more difficult to isolate analysis to a single set of workloads. To combat this, it is recommended to route jobs to relevant subsystems or to assign more descriptive job names with the PASE\_FORK\_JOBNAME environment variable.

## **7.3 CPU Questions**

### **7.3.1 How can I reduce CPU usage on the partition?**

Typically, this requires collecting and analyzing performance data from running jobs. Sometimes you can shift scheduled jobs to run during periods of lower CPU utilization. The best approach depends on which workloads or applications are dominating system performance. See the following sections for tuning specific application types:

Tune SQL performance (see [SQL performance section](#)).

Tune RPG/COBOL application performance (see section on [RPG/COBOL performance](#)).

Tune Java/WebSphere application performance (see [Java/WAS section](#)).

### **7.3.2 What can I do if I can't reduce my CPU consumption?**

- Change shared processor partitions to be uncapped and set virtual processors (VPs) higher than number of configured processing units.
- Don't configure more VPs than are needed for uncapping and workload demand (from spare capacity).
- Don't set the number of VPs higher than the number of licensed processors in the machine (or processors in the shared pool if dedicated processor partitions are not donating unused cycles).
- Don't exceed your software licenses.
- Add more processing capacity if system, database, and application tuning has not reduced CPU usage sufficiently.
- Use powerful performance tools to understand processor utilization and logical partitioning effects (e.g. CPU queuing, latencies, and shared processor behavior). The iDoctor tool is a valuable tool to understand these effects.

### **7.3.3 Why does my partition's CPU utilization show a value greater than 100%?**

Normally a partition's CPU utilization ranges from 0% to 100%. There are several reasons why the CPU utilization would be reported to be greater than 100%:

1. If the partition is configured as shared and uncapped, the CPU utilization can exceed 100%. The amount the partition can exceed 100% depends on multiple settings including the number of physical and virtual processors, the size of the shared processor pool, and the partition's uncapped weight. This occurs when additional available capacity is allocated to the partition.
2. If performance data is refreshed rapidly, performance tools such as WRKACTJOB may not have enough time to gather all the required counter values and the resulting CPU utilization may

inaccurate. As a result, CPU can be incorrectly reported as more than 100%. To improve accuracy, allow some time before refreshing the performance counters.

3. The IBM i operating system provides “helper threads” to assist with functions such as database. If these helper threads consume considerable CPU, there may be intervals in which the partition’s CPU utilization is reported as greater than 100%.

### ***7.3.4 Why doesn't the System CPU utilization equal the sum of the CPU utilization of the individual jobs in WRKACTJOB?***

The CPU utilization information for individual jobs is tracked differently than the overall system CPU utilization. Each process has counters associated with that process to track a wide array of statistics, including CPU. The system utilization comes from counters that do not keep track of each individual process. Each value should only be used to compare to itself - - System CPU in one period to System CPU in another; Job CPU to either the CPU of another job or the job CPU of another period.

Keep in mind that IBM i system tasks are not included in WRKACTJOB command output. They are included in the WRKSYSACT command output.

### ***7.3.5 How can I keep certain workloads from impacting my whole partition?***

To reduce the impact of selected workloads from affecting the entire partition consider the following tips:

- Configure Workload Groups to limit the amount of CPU being used. For more information on Workload Groups, see: [Managing Workload Groups](#)
- Reduce the priority of the jobs and increase the priority of other jobs.
- Put the jobs in their own subsystem to control the amount of memory they can consume. Then configure the shared pool settings properly to limit how large (and small) each memory pool can become.
- Consider limiting which jobs utilize the parallel processing capabilities of Db2 SMP (5770SS1 Product option 26 Db2 Symmetric Multiprocessing), especially for long running SQL queries such as ad-hoc queries or reporting. The PARALLEL\_MIN\_TIME option available in IBM i 7.5 can be used to help focus parallel processing use on the queries where it is most useful. See [New QAQQINI control PARALLEL\\_MIN\\_TIME](#) for more information.
- Collection Services collects data for workload groups. Performance Data Investigator and iDoctor can be used for visualization and analysis of workload groups.

### ***7.3.6 The performance of my system is inconsistent. What could cause that?***

One of the first things to check is your system’s energy management settings. IBM Power processor-based systems can be managed by an HMC. These tools allow users to implement power-capping or power-saving modes. When you use power-capping or power-savings modes and these technologies are in effect, the frequency of the processor may change.

For more information on EnergyScale, see:

[IBM EnergyScale for POWER9 Processor-Based Systems](#)

and

[IBM EnergyScale for Power10 Processor-Based Systems](#)

If the processor is slowed down in order to save energy, that's something that can be important to be aware of if you're concerned about performance. The scaled CPU time indicates whether the energy-management features took effect on that partition.

### **7.3.7 Why do I see CPU queuing time when my CPU utilization is not excessive?**

CPU queuing occurs when more jobs, threads, or tasks want to be dispatched to the CPUs in a partition than there are CPUs currently available to run on. Generally, we see CPU queuing time when CPU utilization is fairly high, and it often shows that there is more work than what the partition processing resources can accommodate.

However, CPU queuing time can occur when CPU utilization is low for other reasons. One reason is that the workload is “bursty”. Each job, threads, or task may not use much processing time, but they are still all attempting to run **at the same time**, and thus some of them have to wait for a CPU to become available. This causes CPU queuing time.

Another reason CPU queuing can occur at low utilizations, is when the workload has many very short requests for CPU. This leads to threads constantly switching in and out of the processor. Every switch into the processor has at least some small amount of CPU queuing time, so if there is a high rate of switching in and out of the processor, this shows up as CPU queuing time. Collection Services reports the average CPU utilization for the entire collection interval, typically set to 5 or 15 minutes. This can mask short bursts of high CPU activity that result in CPU queuing. Using a smaller Collection Services interval, or collecting Job Watcher data, can allow you to better see bursts of high CPU.

Also, if [Workload Groups](#) are implemented, any dispatch latency time that may be introduced by limiting the work to a particular number of cores is reflected as CPU queuing time. Both PDI and iDoctor have graphs available to better understand the latency effects.

### **7.3.8 Why didn't my performance improve when I added cores to my partition?**

Adding cores to a partition should reduce the overall CPU utilization but may not increase performance. There are two main reasons this can happen. The first case is when the workload is not CPU constrained. Meaning the jobs running are mostly waiting on I/O or some other resource rather than CPU. In this situation, the real bottleneck was not addressed. An analysis tool such as iDoctor or PDI can be used to identify the correct bottleneck.

The second case is when there are more cores than active jobs or threads. For example, if your batch process runs by using a single threaded job, it cannot take full advantage of more than one hardware thread in a single core. Changing the workload to use multiple jobs or threads allows more work to be completed in parallel, reducing the overall run time.

### 7.3.9 *Why doesn't the CPU utilization reported by the HMC match the CPU utilization reported by IBM i?*

The HMC and IBM i are measuring utilization differently. IBM i uses CPU utilization to report how much capacity remains in the partition. For example, if the partition is 50% utilized, there is approximately half the capacity of the partition remaining. The HMC is measuring how much of a core's capacity remains for other partitions to use. When a core is dispatched to a partition, the HMC considers it to be 100% busy even if the partition only dispatches work to 1 of the 8 hardware threads on the core. In this example, IBM i would report the core to be around 30% busy vs. the HMC reporting it to be 100% busy.

For a detailed explanation of the differences in CPU reporting, see the following PowerVM blog post: [Understanding Processor Utilization on PowerVM](#)

## 7.4 Memory Questions

### 7.4.1 *How much memory should I configure for my partition?*

The minimum partition memory size is 2GB. However, the recommended minimum partition memory size is 4GB.

#### [Miscellaneous Limits](#)

### 7.4.2 *How should I configure my partition's memory pools?*

Every partition is different, but here are some general guidelines:

- In general, enable Expert Cache on most user pools (i.e. \*CALC). Read more here: [Expert Cache - How It Works](#)
- Add memory to the machine pool if the fault rate of that pool is more than **10 faults per second** and the amount of memory used in the pool is high.
- Set the maximum activity level high enough to avoid transitions to ineligible, but not higher than necessary.

### 7.4.3 *Should I turn on the Performance Adjuster (QPFRADJ)?*

Most customers can benefit from using the performance adjuster. The following experience report is helpful for understanding QPFRADJ: [Performance Adjuster](#).

Make sure you configure the shared pool settings properly to limit how large (and small) each memory pool can become. If you chose not to use QPFRADJ, make sure the max activity level is high enough to avoid transitions to ineligible. Java and Open Source applications have special considerations when it comes to QPFRADJ. See the [Java chapter](#) for details.

### 7.4.4 *How can I tell what is in my partition's memory pools?*

The DMPMEMINF command can provide this information.

This command is described in the IBM Docs:

#### [Dump Main Memory Information \(DMPMEMINF\)](#)

#### **7.4.5 *When should I add more memory to my partition?***

As a general guideline you should limit the amount of time spent waiting on disk faults to 25% of the runtime of an average job on the system. If application optimization cannot reduce jobs below this level, consider adding more memory.

Also, check the performance of the I/O subsystem to ensure disk requests are being completed with good response times.

#### **7.4.6 *What are the "Memory page fault guidelines"?***

With the wide variety of applications that now run on IBM i it is not possible to come up with guidelines that would satisfy every scenario. As a result, IBM no longer publishes general faulting guidelines for user pools.

### **7.5 I/O Questions**

Many of the approaches that reduce CPU usage can also reduce the number of I/O operations, especially those related to improving database and SQL performance. There are also opportunities to improve the performance of the I/O subsystem.

#### **7.5.1 *What are some common storage subsystem configuration changes that can hurt performance?***

- Mixing drive sizes in the same ASP.
- Having some IOAs with more units than other IOAs.
- Having too many units on the same IOA.
- Having a single parity set when utilizing paired (or dual) SAS IOAs.
- Maintaining consistency in drive size helps keep performance consistent. Do not include new drives in existing arrays if your adapter supports this. Instead, stop protection and start protection so all drives contain parity data. This avoids configuring drives with more capacity, which leads to increased IO requests and the potential to be hot.
- Adding new drives to an ASP without using the "Add and Balance" option.
- Reducing the number of drives when migrating to a new storage solution.
- Implementing a high availability replication solution without a properly sized link between source and target storage subsystems.
- Introducing Real-time Compression to an SVC, Storwize, or FlashSystem V840/V9000 storage system. See chapter 7.6 of [Introducing and Implementing IBM FlashSystem V9000](#)
- Utilizing VIOS for storage provides additional functional capabilities but may cause slower performance than dedicated storage options.

#### **7.5.2 *How can I analyze I/O subsystem performance?***

- Use Collection Services and Disk Watcher to collect and analyze disk performance.
- Use the SQL Plan Cache and Visual Explain to identify SQL causing a large number of I/Os.

Read the [RPG/COBOL and Native I/O](#) section for more tips.

#### **7.5.3 *I can't modify my applications. What can I do to improve my I/O performance?***

- Install more memory or disk drives.
- Install SAS Solid State Drives (SSDs) or NVMe devices and move hot objects to them.
- Use the Workload Estimator or contact an IBM Business Partner or Sales Representative to size for a storage solution that delivers faster performance. See section 8.5.6.



#### **7.5.4 *How do I virtualize IBM Serial-Attached SCSI (SAS) adapters for the best performance?***

For the best performance, consider using IBM i as the storage host rather than using a Virtual I/O Server (VIOS).

Section 3.9 titled “VIOS vSCSI disks and IBM i client partitions” of the “IBM Power Systems RAID Solutions Introduction and Technical Overview” Redbook contains special configuration considerations that address the way IBM i requires disk formatting to avoid potential performance problems.

[IBM Power Systems RAID Solutions Introduction and Technical Overview](#)

The following link has even more details on the documentation on the requirements and setup of the performance boost:

[SAS Adapter Performance Boost with VIOS](#)

#### **7.5.5 *How many and what type of storage I/O products meet performance requirements for an upgrade or new system or workload?***

White papers on the Performance management on IBM i Resources website are available for general performance knowledge, already documented comparisons, and tuning suggestions.

- SAP on IBM i Solid State Disk (SSD) Usage Recommendations at:  
[SAP on IBM i Solid State Disk \(SSD\) Usage Recommendations](#)
- IBM Power Systems with IBM i using Solid State Drives to boost your Oracle’s JD Edwards EnterpriseOne performance at:  
[IBM Power Systems with IBM i using Solid State Drives to boost your Oracle’s JD Edwards EnterpriseOne performance](#)

Information about various IBM SAN storage servers that can be used with IBM i can be found on the following website:

<https://www.ibm.com/it-infrastructure/storage>

See appendix A of the "IBM FlashSystem Best Practices and Performance Guidelines" at:

[IBM FlashSystem Best Practices and Performance Guidelines](#)

For particular direct attached storage products:

- Historical information can be seen in versions of the IBM i Performance Capabilities Reference previous to the April 2014 version, at:  
[Performance Capabilities Reference – February 2013](#)
- The most recent comparison of SAS adapter technology can be found at:  
[Performance Study of 2nd Generation IBM Power Systems SAS RAID Adapters Designed for Solid State Storage](#)

#### **7.5.6 *How can I tell if a FlashSystem solution is the best storage configuration for my environment?***

There are many websites that explain the performance, functionality, and cost advantages of an IBM FlashSystem over both hard disk drives (HDD) and solid state drives (SSD). The following website answers many of these questions:

<https://www.ibm.com/flashsystem>

Nevertheless, there can still be conditions that prevent your environment from getting the expected performance improvements when moving to a FlashSystem solution (which can include average read response times as low as 0.5 milliseconds per I/O). One of the most common reasons is that the current IBM Power processor-based system configuration is experiencing some other resource constraint, or wait condition, unrelated to the I/O subsystem. The results of a faster FlashSystem solution could be masked by these conditions and might even make the response times worse if they push more work to drive additional thrashing. It is also possible that write intensive workloads could become slower. Even though Flash has acceptable write response times, nothing can compare to the speed of writing to cache in an IOA. Internal drives are typically configured to IOA's with very large write cache allowing most write IOP's to be asynchronous (or appear to be so due to fast response times).

To help customers identify situations that could prevent the unacceptable response times, and also to help project the run time improvements of the most I/O intensive jobs, IBM Technology Expert Labs offers an "IBM i Workload Assessment for FlashSystem Solution". For more information, complete this form:

[IBM Technology Expert Labs Contact Form](#)

Also, note that the SSIC (System Storage Interoperability Center) states the following regarding 4096 sector size used for natively attached FlashSystems. Also reference section 8.5.9 regarding the performance implications of 4096 sector drives on IBM i.

Certain unique workloads show lower than expected performance for IBM i hosts when attached to IBM FlashSystem 840/900 storage without SAN Volume Controller (SVC). It is recommended these opportunities be vetted with a PoC to ensure the clients' workload does not encounter this performance issue. In addition, IBM i is maintaining a list of recommended PTFs for any clients running in a 4096 sector size configuration which is required with an attached FlashSystem 840/900. This list can be found here:

<http://www-01.ibm.com/support/docview.wss?uid=nas8N1020957>

### ***7.5.7 How can I tune my Virtual Fibre Channel attached external storage?***

Here is a list of recommendations that can be used to ensure vFC/NPIV environments perform as well as possible when connected to supported External Storage Systems:

<https://www.ibm.com/support/pages/ibm-i-virtual-fibre-channel-performance-best-practices>

To take greater advantage of the high command throughput rates of vFC attached solid state storage, 7.5 TR1, 7.4 TR6 and 7.3 TR12 utilize a multiple queue (MQ) feature that was added to VIOS version 3.1.2. The blog titled [Modernization of PowerVM NPIV stack: from Single to Multiple Queue Support \(ibm.com\)](#) provides a description of MQ and some impacts observed with AIX as a client partition. The increased parallelism MQ provides to an IBM i partition can increase command throughput for workloads that need high io/sec and have relatively short (<16KB/io) average storage io lengths.

### ***7.5.8 How do I analyze I/O performance if I'm using external storage?***

There are many components to consider when external storage is being used. Latency can be added in the virtualization layer (VIOS and Power Hypervisor) and SAN which are located between the IBM i and storage. A good strategy is to compare the IBM i drive service times with the response times measured

on the storage. If the service times and response times are close to the same, then the problem is likely in the storage. If the IBM i service time is higher than the storage, then latency is likely being added in the SAN or virtualization layer. Collection Services can be used to measure the service times on the IBM i. There are several ways to measure the response times on the storage depending on what is being used.

IBM iDoctor for i – Collection Services Investigator can be used to analyze data that is automatically collected from attached DS8000 Storage systems. More information can be found at this link:

[iDoctor Documentation](#)

IBM Storage Insights provides an unparalleled level of visibility across your storage environment to help you manage complex storage infrastructures and make cost-saving decisions. Benefits include enhanced performance monitoring, capacity planning and capability to troubleshoot issues faster. Two editions are available, including a no charge offer to monitor the basic health of your storage. More information can be found at this link:

[IBM Storage Insights](#)

The Storage Monitoring feature in IBM Spectrum Control (formerly known as IBM Tivoli Storage Productivity Centre (TPC)) can be used to analyze performance of most IBM Storage Systems. More information can be found at this link:

[IBM Spectrum Control](#)

### **7.5.9 What are the IBM i performance effects of using 4096 byte sector drives?**

There are options for 4096 byte formatted internal HDDs and SSDs both natively and virtualized through VSCSI from VIOS or an iHost. Flash Systems external storage are also formatted for 4096 byte sectors. Ensure the latest relevant PTFs are applied before implementation. The IBM i 7.2 and 7.3 list can be found here:

[PTF listing for 4096 disk sector support](#)

If implementing 4096 byte sectors, there is additional overhead associated with storage management cleanup tasks. In order to lessen the impact, review the workload for inefficient application issues, specifically high rates of database full opens and high rates of activation group creates and destroys. These are discussed in section 8.16. Consider engaging IBM Technology Expert Labs for assistance in reducing open rates before implementing 4096 byte sectors. Additionally, review the following document regarding the SMFSCLEAN\* tasks.

[SMFSCLEAN\\* tasks and machine level gating in SMFREESPACEMAP4K](#)

### **7.5.10 Should I utilize NVMe drives?**

IBM i 7.5, 7.4, 7.3 and 7.2 various Technology Refreshes support NVMe devices that are available in multiple capacity points, multiple form factors, with various speeds, for enterprise workloads on selected Power servers with Power technology. The IBM i I/O Support Summary [web page](#) describes what releases support which devices and under what configurations (native, VIOS and iVirt). These low

latency devices can be used as IBM i load sources and are able to provide a high number of io/s and enhanced virtualization capabilities.

The PCIe gen4 devices can also provide a significant amount of data throughput (GB/s). Mirroring is required. Certain Power Servers can include only Add-In-Card (AIC) form factor NVMe devices while some Power Servers can accept both AIC and U.2 form factor devices.

NVMe devices have the ability to be virtualized into multiple namespaces. Each namespace is treated as a storage device by Storage Management. It is recommended to allocate as many namespaces as SAS or FC LUNs deemed to be sufficient to meet application throughput requirements. See the NVMe - IBM Documentation for directions on how to configure & create namespaces.

If more space, or more throughput, is required from NVMe devices than can be installed within CECs slots, a NED24 NVMe Expansion Drawer (feature ESR0) that fits into the identical rack space as the latest SAS remote I/O enclosures can be used to house up to 24 U.2 NVMe SSDs. Each ESR0 can provide up to 10x more data throughput (GB/s) than a comparable ESLS SAS drawer and up to 3-4X more command throughput (io/s).

## 7.6 Processor Virtualization Questions

### 7.6.1 *How do I determine if my workload is being impacted by processing capacity entitlement delay?*

A shared processor LPAR is configured with a number of virtual processors and a processor capacity entitlement. When executing instructions, a virtual processor is assigned a physical processor of the shared processor pool, and processing capacity usage is charged to the LPAR at a rate of 1.0 unit of capacity per elapsed time. PowerVM meters the processing capacity usage of a LPAR's virtual processors over a 10 millisecond window, so that over the long term, the partition is guaranteed access to its entitled capacity. When a capped LPAR has consumed its capacity entitlement within a window, its virtual processors are preempted and queued until the next window, at which time the capacity entitlement is replenished. If an uncapped LPAR exhausts entitlement, it may continue to claim a share of unused capacity from the shared processor pool, so its entitlement delay can be reduced. Field SYPTREADY in Collection Services file QAPMSYSTEM records the entitlement delay for a shared processor LPAR, and field JBVPDLY in Collection Services file QAPMJOBMI records the combined entitlement and virtual processor dispatch delays at the job-level.

### 7.6.2 *How can processing capacity entitlement delays be reduced?*

Partition-level processing capacity entitlement delays are indicated by field SYPTREADY in Collection Services file QAPMSYSTEM, and job-level virtual processor delays are indicated by field JBVPDLY in Collection Services file QAPMJOBMI. Entitlement delays can be reduced by increasing the ratio of processing capacity units to virtual processors in the LPAR configuration. This is typically accomplished by increasing the processing capacity units or by reducing the virtual processors, or both.

Figures 14 and 15 illustrate the relationship between entitlement delay and the ratio of processing capacity units to virtual processors in a LPAR configuration. In this example, the workload primarily consists of a single-threaded CPU-intensive SAVLIB operation, with the LPAR configuration being 1 processor, 0.1 processor units (capped) until 15:41:00, when the processor units was increased to 1.0. In this example, notice that the Entitlement Used from HMC viewpoint (%) indicates that the LPAR is consuming its entire entitlement both before and after the change, and that the entitlement delay (CPU

thread ready wait time) at partition-level and processor virtualization delay (Dispatched CPU wait) at job-level are mostly eliminated by the change. This example also illustrates how processing capacity entitlement can be entirely consumed despite the partition CPU utilization being much lower, owing the relative inactivity of the processor's other SMT threads.

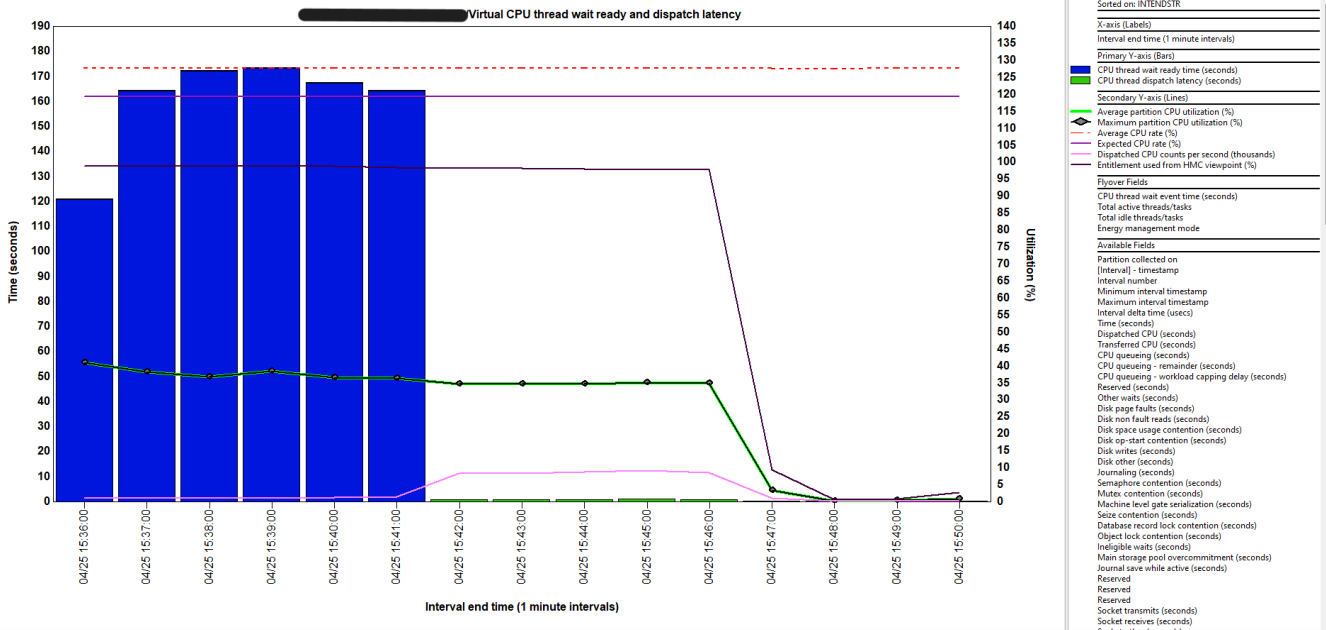


Figure 15 - Partition-level illustration of entitlement delay reduction - IBM iDoctor for i

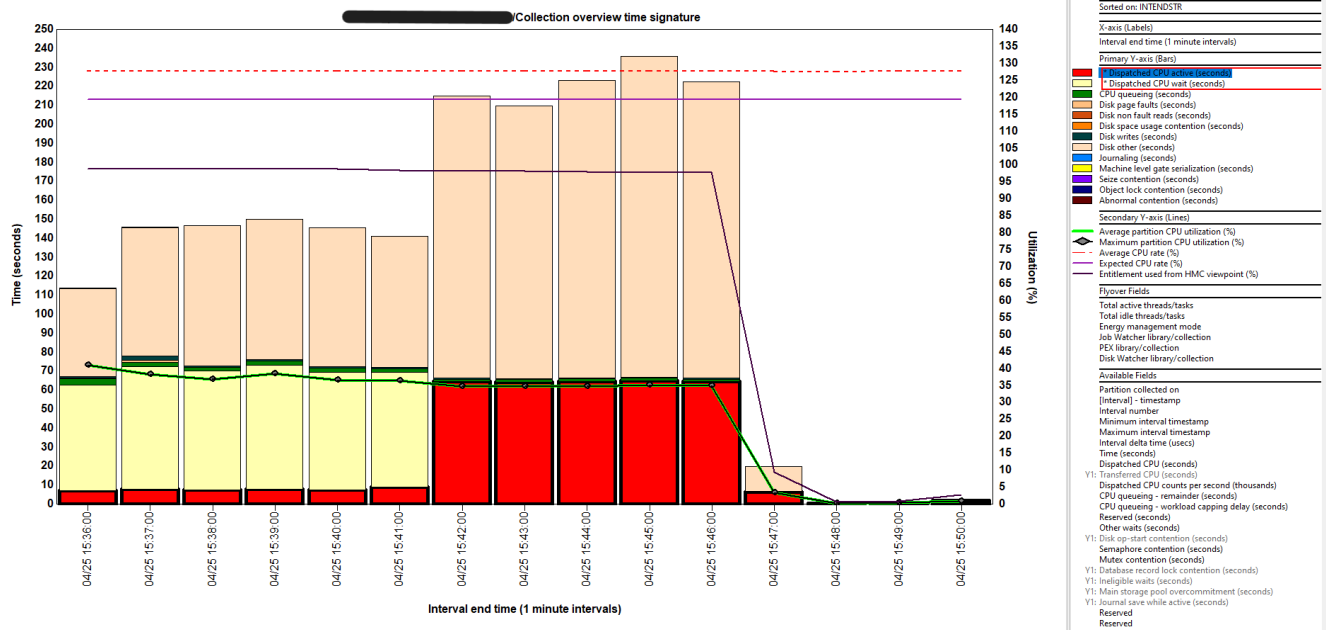


Figure 16 - Job-level illustration of processor virtualization delay reduction - IBM iDoctor for i

If a shared processor LPAR is capped or soft-capped (uncapped with weight 0), then changing it to uncapped with non-zero weight can also reduce entitlement delays, with the extent of relief being dependent on the availability of unused capacity in the shared processor pool.

### 7.6.3 *How do I determine if my workload is being impacted by over-commit of the shared processor pool?*

When actively executing instructions, a virtual processor is assigned a physical processor of the shared processor pool, and all threads of the processor are reserved for exclusive use by the LPAR to which it is assigned. When the number of virtual processors demanded (and not waiting for entitlement) by the shared processor pool's LPARs exceeds the number of physical processors assigned to the pool, then some virtual processors will temporarily experience virtual processor dispatch latency. Field SYPTLATEN in Collection Services file QAPMSYSTEM records the dispatch latency delays for a shared processor LPAR, and field JBVPDLY in Collection Services file QAPMJOBMI records the combined entitlement and virtual processor dispatch delays at the job-level

### 7.6.4 *How can virtual processor dispatch latency delays be reduced?*

This is a challenging question from the LPAR perspective because numerous factors impact virtual processor dispatch latency, and in many cases, these factors are beyond the control of a single LPAR. Specifically, the pattern of virtual processor demands across *all* of the shared processor pool's LPARs determines the virtual processor dispatch latency. The short answer is to reduce the ratio of virtual processors to physical processors of the shared processor pool. This can be accomplished by reducing the number of virtual processors, or by allocating additional capacity to the shared processor pool, or by configuring dedicated processor LPARs to temporarily donate their inactive or idle processors to the shared processor pool. The ratio can also be optimized by observing best practices regarding the configuration of shared processor LPARs, but this discipline needs to be employed across all LPARs for best results. OS-level scheduling behavior and features such as processor folding can impact a LPAR's efficiency with regard to virtual processor demands. In some cases, the problem may require higher-level coordination across LPARs, such as sequencing peak-usage periods.

Figures 16 and 17 illustrate the complex relationship between processor virtualization delay from a single LPAR's perspective and *other* activity in the LPAR's shared processor pool. The shared processor pool has 22 processors initially, with 2 shared processor LPARs of 22 processors, 11 units (uncapped) configured. The LPAR under consideration (LPAR 1) is running COPR, which is the OLTP workload used to establish CPW ratings of IBM Power system models running the IBM i operating system. The approximate timeline of actions is as follows:

14:50 – 15:00 : LPAR 1 is running a COPR workload. LPAR 2 is idle.

During this period, LPAR 1 benefits greatly from idle capacity in the shared processor pool, driving average partition CPU utilization to 180% while incurring minimal processor virtualization delays.

15:00 – 15:20 : LPAR 1 and LPAR 2 are running a COPR workload.

With LPAR 2 also running a COPR workload, the processor pool is heavily over-committed, with 44 virtual processors sharing 22 physical processors. Processor virtualization delays at the system-level and job-level are significantly impacting workload performance in both LPARs.

15:20 : Processor pool expands by 11 processors from 22 to 33.

A dedicated processor LPAR has deactivated 11 of its processors, causing the 11 processors to be available to the partitions of the shared processor pool. The additional processors has reduced the degree of over-commitment, with 44 virtual processors sharing 33 physical

processors. The additional processing capacity is used by the uncapped LPARs, resulting in a reduction of system-level and job-level virtualization delays.

15:40 : Processor pool expands by 10 processors from 33 to 43.

Additional processors were made available to the shared processor pool, to the point where the pool is barely over-committed with 44 virtual processors sharing 43 physical processors. Workload performance in LPAR 1 has nearly returned to the level of interval 14:50 – 15:00, where LPAR 2 was idle.

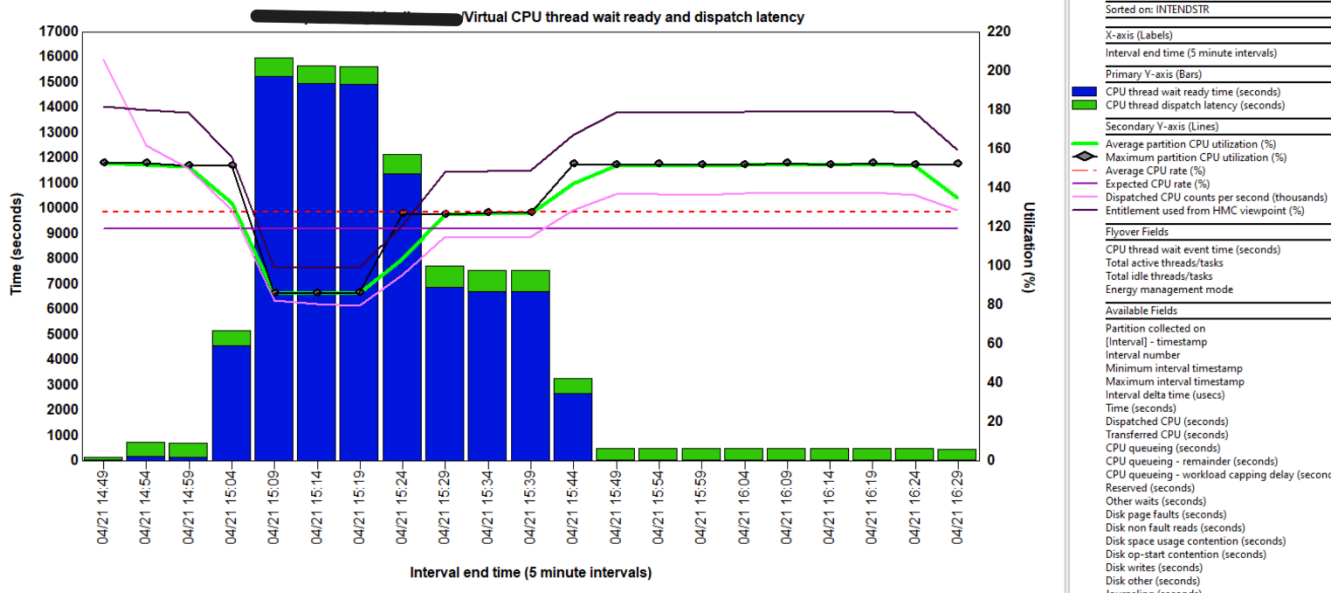


Figure 17 - Partition-level illustration of processor virtualization delay - IBM iDoctor for i.

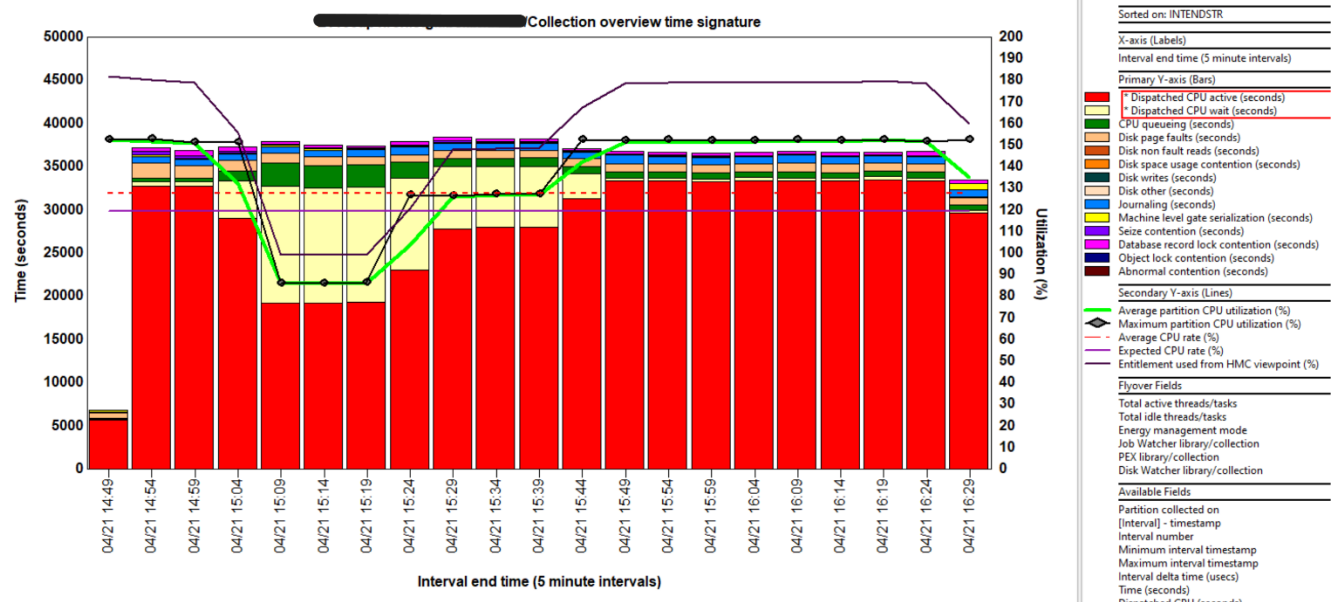


Figure 18 - Job-level illustration of processor virtualization delay - IBM iDoctor for i.

## 7.7 How do I tune IBM i database performance?

See [chapter 9](#) for details on database performance tuning.

## 7.8 How can I improve backup and recovery times?

See the following information to understand how to improve backup and recovery performance:

[Backup and recovery frequently asked questions](#)

For improved IFS SAV performance information:

[Achieve faster IFS save times by using SAV with ASYNCRING](#)

## 7.9 How can I improve IPL time?

Although IPL duration is highly dependent on hardware and software configuration, there are tasks that can be performed to reduce the amount of time required for the system to perform an IPL. The [Experience Report: Reducing System IPL Time](#) report contains many suggestions for reducing IPL time.

Some more recent suggestions to consider are:

- IBM i 7.3 includes a design change to the IPL process to help reduce IPL time for partitions with very large amounts of memory. This improvement is not available for earlier releases.
- The performance of the I/O subsystem can be an important component of IPL time. Customers who upgraded their storage from HDDs to SSDs or Flash storage saw substantial improvements to IPL time. However, not all systems will see improved IPL times through I/O upgrades due to other factors.
- Internal measurements show that moving only the load source from an HDD to SSD does not significantly improve IPL time.

## 7.10 What are considerations for Main Store Dump (MSD) IPL performance?

The “Best Practices for Managing Time Needed for Main Storage Dump (MSD)” Technote has the latest recommendations, enhancements, and PTFs to help manage and reduce time required for MSD:

[Best Practices for Managing Time Needed for Main Storage Dump \(MSD\)](#)

## 7.11 How do I tune journal and recovery performance?

- For IBM i releases before 7.1 make sure that the journal threshold for all user journals is at least 6.4GB. As of IBM i 7.1 this is no longer necessary.

The journal receiver size options (RCVSIZOPT) should preferably specify at least \*MAXOPT2 and \*RMVINTENT.

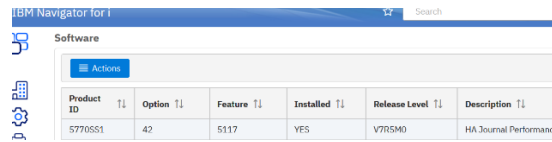
The Journal Management guide has more details on these commands:

[Journal Management Guide](#)

- Consider enabling IBM i Journal Cache. Journal caching is a feature that, per journal, allows the system to bundle or cache journal entries in main storage before writing them to disk. Journal caching modifies the behavior of traditional non-cached journaling in batch. Without journal caching, a batch job waits for each new journal entry to be written to disk. Journal caching



allows most operations to no longer be held up waiting for synchronous disk writes to the journal receiver. Journal Caching (HA Journal Performance) is Option 42 of the IBM i operating system.



The screenshot shows the IBM Navigator for i interface. The main content area is titled 'Software' and contains a table with the following data:

Product ID	Option	Feature	Installed	Release Level	Description
5770551	42	5117	YES	V7R5M0	HA Journal Performance

Journal caching was previously a chargeable feature, but as of June 2022, it is no-charge option. It is distributed with IBM i 7.5 but will have to be downloaded and installed on versions 7.4 and 7.3.

Performance results of using Journal Cache will vary. Batch applications which perform large numbers of changes to the data portion of the journaled objects often see the most performance benefit of journal caching. Applications using commitment control will see less improvement as commitment control already performs some journal caching.

You can leverage the power of IBM i Wait Accounting technology in Collection Services (and Job Watcher) to get an idea whether Journal wait time is an issue. The Journal wait time bucket can be viewed at the partition level, as well as an individual job level. Use the wait graphs in either Navigator PDI or iDoctor to view the data.

For additional usage information and considerations, refer to:

<https://www.ibm.com/docs/en/i/7.5?topic=journals-journal-cache>

<https://www.ibm.com/support/pages/basics-journal-cache>

### 7.12 How do I tune access path recovery performance?

System-Managed Access Path Protection (SMAPP) offers system monitoring of potential access path rebuild time and automatically starts and stops journaling of system selected access paths. In the unlikely event of an abnormal IPL, this allows for faster access path recovery time.

An estimation of how long access path recovery takes is provided by SMAPP, and SMAPP provides a setting for the acceptable length of recovery. For most customers, the default value minimizes the performance impact, while at the same time provide a reasonable and predictable recovery time and protection for critical access paths. But the overhead of SMAPP varies from system to system and application to application.

As the target access path recovery time is lowered, the performance impact from SMAPP increases as the SMAPP background tasks have to work harder to meet this target. There is a balance of recovery time requirements vs. the system resources required by SMAPP. The access path recovery time should **not** be set to \*NONE or \*OFF.

Although SMAPP may start journaling access paths, it is recommended that the most important/large/critical/performance sensitive access paths be journaled explicitly with STRJRNAP.

This eliminates the extra overhead of SMAPP evaluating these access paths and implicitly starting journaling for the same access path day after day.

There are 3 sets of tasks which do the SMAPP work. These tasks work in the background at low priority to minimize the impact of SMAPP on system performance. The tasks are as follows:

- JO-EVALUATE-TSKn (where n is 0 or 1) - Evaluates indexes, estimates rebuild time for an index, and may start or stop implicit journaling of an index.
- JO-TUNING-TASK - Periodically wakes up to consider where the user recovery threshold is set and manages which indexes should be implicitly journaled.
- JO-RECRA-Dx-xx-x and JORECRA-USR-xxx tasks are the worker tasks which sweep aged journal pages from main memory to minimize the amount of recovery needed during IPL.

Here are guidelines for lowering the amount of work for each of these tasks:

- If the JO-TUNING-TASK seems busy, you may want to increase SMAPP recovery target time.
- If the JO-EVALUATE task seems busy, explicitly journaling the largest access paths may help. Also, look for jobs that are opening/closing files repeatedly.
- If the JORECRA tasks seem busy, you may want to increase journal recovery ratio.
- Also, if the target recovery time is not being met there may be SMAPP ineligible access paths. These should be modified to become SMAPP eligible. You can use the Display Recovery for Access Paths (DSPRCYAP) command to see a list of access paths that are not eligible for SMAPP.

To monitor the performance impacts of SMAPP there are Performance Explorer trace points and a substantial set of Collection Services counters which provide information on the SMAPP work.

For more information on reducing access path recovery time, see the IBM Docs entry at:

[Reducing time in access path recovery](#)

### 7.13 How do I tune network performance?

The network adapters being used determine the maximum speed which could be reached, e.g. 1Gb, 10Gb, etc. (at least in theory). However, the protocol being used, the networking parameters set and the quality of the connections as well as other systems in the same subnet in the network determines the actual performance in terms of network throughput and speed. Performance Collection Services collect vital data from the use of networking resources and can be used to determine the actual speed and throughput numbers of the network.

- Consider increasing the TCP/IP buffer for send and receive operations via the CHGTCPA command to a value greater than the default of 64KB. This can significantly reduce the network traffic.
- Ensure the parameters for current line speed reflect what the adapter is being capable of, e.g. 1Gb, 10Gb, etc. A single device with a lower line speed capability will force the whole subnet in the network to run at the lower speed and can severely degrade network performance.
- Ensure the current DUPLEX parameter is set to \*FULL so the connection can be used for send and receive at the same time.
- Consider increasing the maximum frame size from 1496 bytes to 8996 bytes as this can significantly reduce the traffic between the system and the next network router and speed up

- the connections especially when virtual Ethernet connections are used.
- Consider using multiple network connections together and use link aggregation to bundle as shown in the following diagram for example four 1Gb connections into one 4Gb link. This aggregated link provides a significant bandwidth improvement.

For more information, see the IBM Docs section:

[Ethernet Link Aggregation](#)

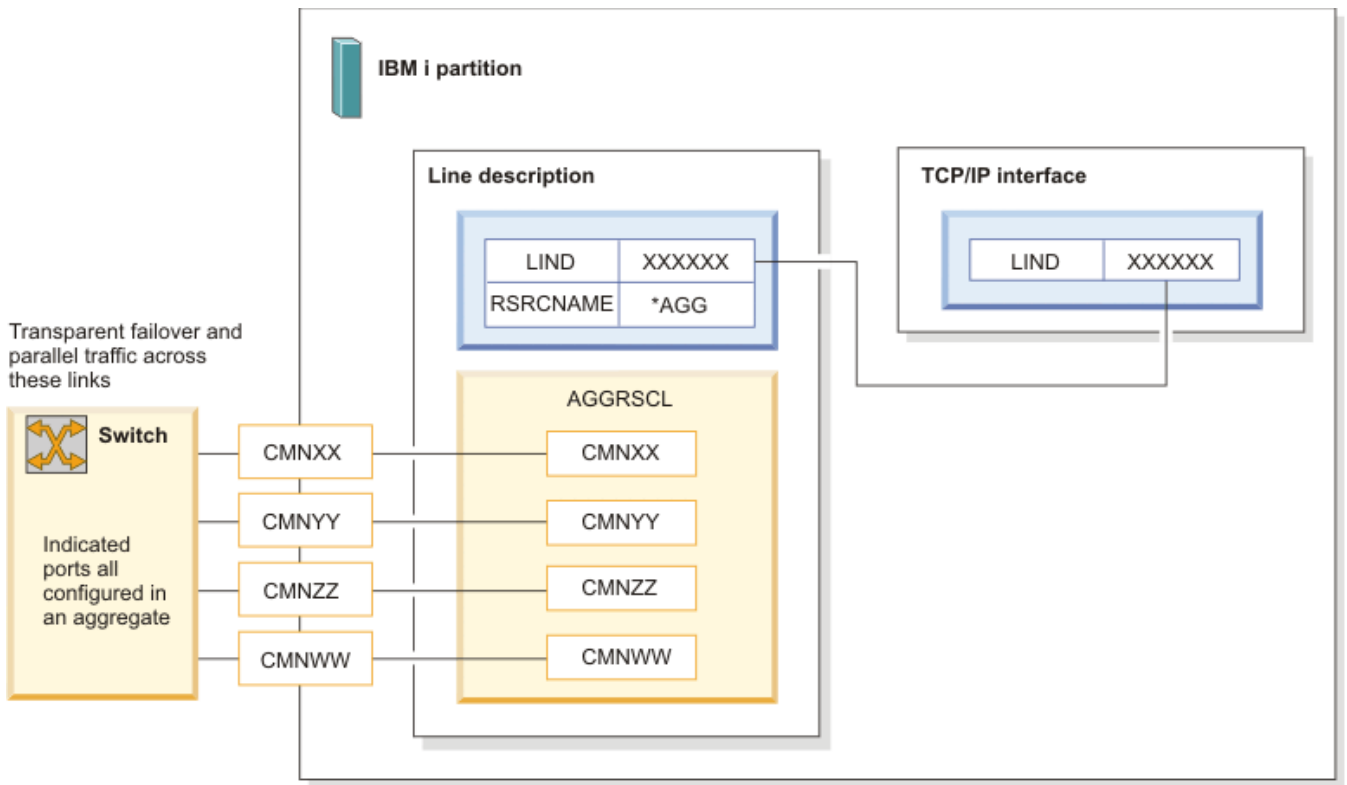


Figure 19 - Network Link Aggregation

More information on optimizing communications performance can be found in the following IBM Docs section:

[Optimizing communications performance](#)

### 7.14 How can I improve the performance of Java and/or WebSphere Application Server applications?

See the [Java and WebSphere chapter](#).

### 7.15 How do I tune RPG/COBOL application performance and native I/O file access?

See the [RPG/COBOL and native I/O chapter](#).

### 7.16 Can security settings affect performance?

Protecting information is an important part of most applications. Security should be considered, along with other requirements, at the time the application is designed. When security is integrated properly

into application design, it will result in the best performance. It is when security is not implemented correctly that performance problems typically occur.

Defining a large number of private authorities can negatively affect performance. The following links to IBM Docs show how authority checking is done. A properly architected solution will balance security concerns with performance.

[Flowchart 1: Main authority checking process](#)

[Flowchart 2: Fast path for object authority checking](#)

When implemented correctly, the application fast path will generally be taken through program adoption of the owner. When public and private authorities are added, complications can arise, impacting performance.

Collection Services includes two metrics that can be useful for monitoring authority activity. The metrics are SYAUTH and SYNUAL in the file QAPMSYSTEM. SYAUTH counts the number of object authority checks system wide. SYNUAL counts the number of non-cached user authority lookups.

## 7.17 What are some inefficient application issues I can identify at a system level?

### 7.17.1 Look for high rates of database full opens

Having a high rate of database file full opens can lead to poor performance. Full opens can come from SQL or programs using native I/O. Collection Services collects the number of database file full opens at both a system level and a thread level. This data can be visualized using either PDI or iDoctor.

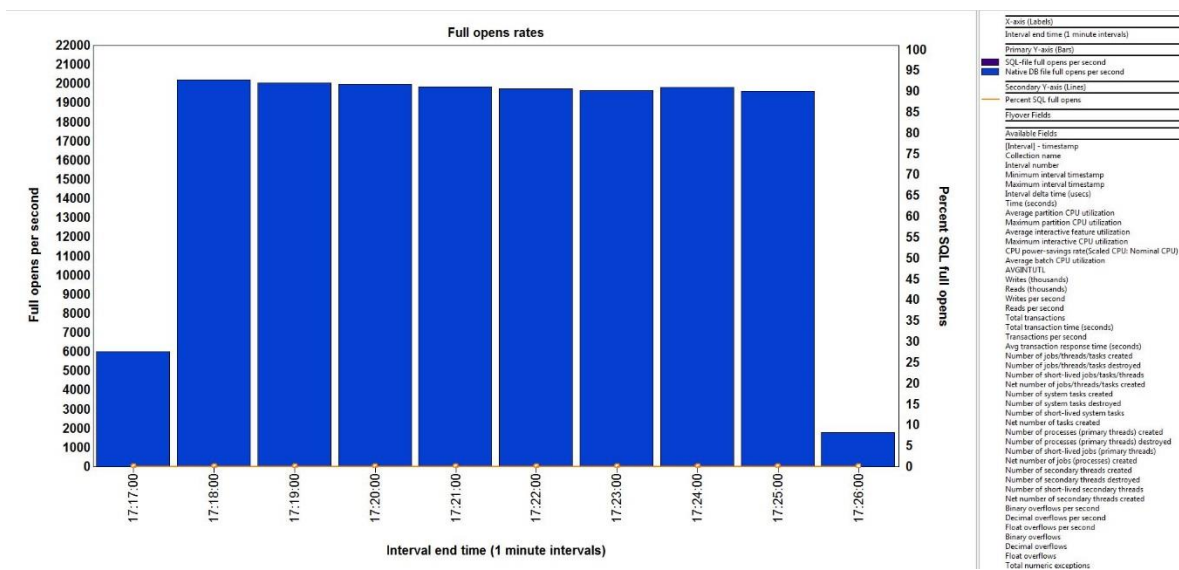


Figure 20 - Full Opens in iDoctor Collection Services Investigator

To identify the programs causing the full opens, a PEX trace can be collected using the \*DBOPEN OS event. The PEX trace can be analyzed using the PEX Analyzer component of iDoctor

A high rate of database file full opens is a common cause for poor performance on IBM i partitions. Full file opens are heavy-weight operations that should be minimized. It is not uncommon for jobs with high

full open rates to be wasting one third or more of their CPU time doing full file opens, when they typically don't need to.

A general guideline for partition wide full open rates is less than 1000 per second on small to medium sized partitions, and low single digit thousands per second on large partitions. Additionally, keep the full open rate to well under one hundred opens per second for any single job. Regardless of the size of the partition, the lower the number the better for performance and scalability.

These full open guidelines are independent of disk storage technology.

### 7.17.2 Look for high rates of activation group creates/destroys

Having a high rate of activation group creations and destroys is also inefficient. Starting with IBM i 7.2, Collection Services collects the number of groups and programs activated at a system and thread level. This data can be visualized using iDoctor.

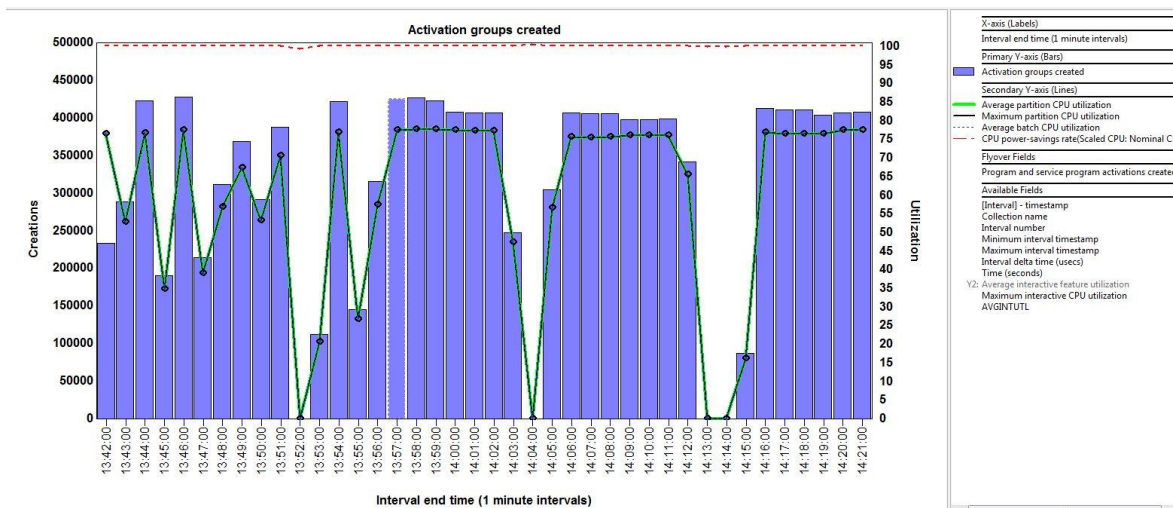


Figure 21 - Activation Groups Created in iDoctor Collection Services Investigator

To identify the programs causing the activation group creates, a PEX trace can be collected using the \*ACTGRPCRT Base event. The PEX trace can be analyzed using the PEX Analyzer component of iDoctor.

### 7.18 What is the largest number of cores that can be assigned to an IBM i partition?

There are two partition core limits for IBM i. The first is the standard limit which is 48 cores for all supported IBM i releases. Customers who need to exceed this limit can go through a no-charge assessment process to validate that their partitions will benefit from having additional cores. The maximum number of cores allowed depends on the level of IBM i and Power hardware, as well as the results of the assessment.

Consult the following matrix showing the maximum single-partition core limits. For more information, contact Eric Barsness ([ericbar@us.ibm.com](mailto:ericbar@us.ibm.com)) or Mike Gordon ([mgordo@us.ibm.com](mailto:mgordo@us.ibm.com)).

OS level	Max Allowed w/Assessment		
	Power8	Power9	Power10
IBM i 7.2	96	96	96
IBM i 7.3	96 SMT8	96 SMT8	96 SMT8
	192 SMT4	192 SMT4	192 SMT4
IBM i 7.4	96 SMT8	192	192
	192 SMT4		
IBM i 7.5	-	192	240

Figure 22 - Maximum Cores in Partition

### 7.19 How can I improve the scalability of my large IBM i partition?

Something that is done inefficiently but infrequently may not be a problem. If that activity is done more and more frequently it can become a major problem. It is not uncommon for a program written 10 years ago to access a small file accessed by a few users be an issue today because that file has grown to millions of records and is being accessed by hundreds of users. To avoid such a problem, make sure your applications are not doing the following "worst practices":

- issuing a high rate of full file opens and closes
- creating and destroying activation groups at a high rate
- frequently destroying large activation groups
- causing a high rate of job initiations and terminations

The rate metrics for these scaling issues are collected by Collection Services and can be viewed using either Performance Data Investigator (PDI) or iDoctor Collection Services Investigator (CSI). See [section 8.16](#) for more information on some of these metrics.

In addition to application coding practices, the following partition configuration areas are important for scalability:

- partitions with many cores should use dedicated processors for best performance
- placement of partition resources (memory and cores) on the physical hardware

Chapter 6 of the [IBM POWER Virtualization Best Practices Guide](#) provides information on how to improve partition placement.

## 7.20 How can I proactively ensure that batch performance will improve with my system upgrade?

When upgrading to a newer processor model or feature, batch performance may not be improved unless some upfront work is done to determine what limitations may be currently constraining the batch jobs.

Here are some considerations for batch performance:

1. Determine if there are application constraints such as locking on an object or resource. Proper wait time analysis should be done to identify the bottleneck and how to best resolve it. See the [Components of Response Time section](#) earlier in this document.
2. Determine if storage I/O response times will be improved with the new system. When systems are upgraded, often the storage I/O is migrated as well. If this is the case, I/O response times will be the same or worse due to heavier stress with faster processor/memory. When moving to new storage subsystems, consider I/O response time differences and cache differences. The I/O response time can be analyzed with tools such as Collection Services.
3. Determine if memory constraints are not tightened due to more work on the system. Memory performance can be analyzed with tools such as Collection Services.
4. Determine if the batch workload is truly single-threaded. If this is the case, simply adding more cores to the partition (or just a higher system CPW rating) may not help. Faster processor frequency is the key to improving many well-behaved single-threaded applications. Better yet, overlapping work by utilizing multi-processing or multiple jobs may help even more.
5. IBM Technology Expert Labs can assist with analysis of batch jobs and help to reduce batch window processing times.

## 7.21 How can I check the “Interactive Capacity” setting on my system?

Some IBM i systems allow Interactive Capacity to be limited. This reduces the performance for non-batch jobs (also known as 5250 interactive jobs).

To avoid this limitation, the '100% 5250' feature must be enabled.

Without the '100% 5250 feature', a CPI1479 message will appear in the QSYSOPR message queue and interactive performance will be poor while batch jobs remain unaffected. The Interactive Capacity values shown below will be '0'.

Interactive Capacity values are stored in the Collection Services QAPMCONF file and can be easily viewed using either Performance Data Investigator or iDoctor.

To view in Performance Data Investigator, display the QAPMCONF perspective for the Collection Services collection of interest.

Interactive Limit (%):	100
Time Interval (seconds):	300
Interactive Threshold (%):	100

To view in iDoctor: launch Collection Services Investigator, then right on the collection of interest and select Properties.

Interactive limit	100%
Interactive threshold	100%

You can also view in ASMI by opening the “On Demand Utilities” menu on the left column. Then click “CoD VET Information” below that. Interactive capacity is listed as item 11 on this list.

VET Capability Settings	
Capability bit information	Capability bit status/count
0 AIX allowed (license needed)	False
1 AIX allowed (license not needed)	True
2 RPA I/O hosting	True
3 AIX diagnostics	True
4 Subprocessor partitioning	True
5 LPAR creation allowed	True
6 Greater than 128 processors per partition	True
7 CAPI	True
8 i5/OS partitions allowed	True
9 Reserved	True
10 Live partition mobility	True
11 i5/OS 100% interactive capacity	True
12 Active Memory Expansion	True
13 Reserved	True
14 NEBS enabled	False

Figure 23 - Interactive Capacity

If this option is set to False, your interactive/5250 session performance will be affected.

See the following website for more details:

[Power9 IBM i P20 \(and greater\) systems 5250 interactive performance and CPI1479](#)

## 7.22 What are considerations for IBM Power Virtual Server partitions

IBM i supports IBM Power Virtual Servers and can be configured much like a normal partition. Larger system configuration options such as IBM Power System E880 and IBM Power System E980 and



smaller system configurations such as IBM Power System S922 are provided.

Processors can be configured as dedicated or shared. Dedicated processors provide the most consistent processing performance. For more tips on virtualization performance, see [Virtualization Best Practices](#).

IBM Power Virtual Servers provide two storage options: Tier 1 and Tier 3. The **Tier 1** storage type is best for customers who require higher storage throughput. Customers who do not require high storage throughput and are looking to minimize costs can select **Tier 3**. The storage types cannot be changed once the volume is created. A virtual machine (VM) cannot have disks from both storage types.

The storage tiers in Power Virtual Server are based on I/O operations per second (IOPS). The performance of your storage volumes is limited to the maximum number of IOPS based on volume size and storage tier. Although, the exact numbers might change over time, the **Tier 3** storage is currently set to 3 IOPS/GB, and the **Tier 1** storage is currently set to 10 IOPS/GB. For example, a 100 GB Tier 3 storage volume can receive up to 300 IOPS, and a 100 GB Tier 1 storage volume can receive up to 1000 IOPS. For more IOPS for a volume you can either increase the size of the volume or choose a tier that provides more IOPS/GB. In some cases, it may be more cost efficient to increase the size and use Tier 3 than using a smaller size volume and using Tier 1. After the IOPS limit is reached for the storage volume, the I/O latency increases. Even brief bursts in IOPS that exceeds the configured Tier throughput limit can trigger IOs with much higher latency that can impact application response times.

While not unique to this environment, consideration should also be given to the network. Be sure to understand how a change in network latency between the client and server may affect client-server application performance. While client-server applications designed requiring a high volume of network traffic may perform well when both the client and server are physically close together, application changes can be required as additional network latency is introduced.

It is important to properly size your partition before you configure your virtual server partition. Ensure that the processor configuration can meet your peak performance needs. Closely analyze your storage performance needs before choosing Tier 3. If you are not confident that your peak storage needs can be satisfied with the Tier 3 performance offering, choose Tier 1 instead. Be sure to size your partition processing and storage performance capacity for peak throughput that you may experience during the year.

### 7.23 What are some considerations for use of data compression in IBM i?

IBM i 7.5 adds support for compressing data using the DEFLATE compression format (which is the underlying format for zip files, gzip, and the C zlib library). That support is available as a new compression format option on the SAVLIB command's DTACPR parameter (\*ZLIB) and a new format option for the CPRDATA MI instruction. In addition, an IBM i partition running on a Power 10 system can use the Power processor's Nest Accelerator (NX) to accelerate handling of DEFLATE data in those interfaces.

The performance of any compression algorithm varies drastically based on the nature of the data that is being compressed. All data compression works by reducing the redundancy in the input; data that is completely unpredictable cannot be compressed, and data with predictable patterns can be

compressed very well. DEFLATE compressors will also vary in their throughput depending on the input data.

On the data that IBM i has tested, IBM i's DEFLATE implementation typically compresses better than LZ1 (which is the compression offered by SAVLIB DTACPR(\*HIGH)); that is, DEFLATE typically eliminates more redundancy, leading to smaller output than LZ1. If NX support is not available, DEFLATE typically gives similar CPU and wall-clock-time throughput to LZ1. When NX is used, DEFLATE compression operations typically offer at least 5 times faster throughput than LZ1.

Given that, there are significant performance gains to be had by ensuring that a partition has access to NX resources. There is a limited amount of NX capacity available system-wide for DEFLATE operations, which is shared by all of the partitions in the system. Partitions can reserve NX resources ("Quality of Service credits") for DEFLATE/gzip from the Hardware Management Console; each credit is the capability to have one request outstanding to an NX. Any credits reserved this way reduce the number of requests that other partitions can have outstanding. The credits that are not reserved by any partition are shared by any partitions that attempt to send more requests at a time than they have reserved.

To ensure NX resources are available to your partition, ensure that your partition is in Power 10 compatibility mode, and reserve credits at the HMC partition General Properties, Advanced section, Supported Hardware Accelerator Types. For more information on devices that can be used for IBM i's DEFLATE implementation, refer to sections 8.22a and 8.22b.

**A) How do I maximize data compression performance when saving to \*SAVF files?**

[Section 7.23](#) outlines IBM i's DEFLATE implementation and its performance benefits on P10. DEFLATE can be used for saving to \*SAVF. In addition to DEFLATE (DTACPR (\*ZLIB)), there is support for \*MEDIUM, \*HIGH, \*LOW, and \*NO compression as well. For more information on creating \*SAVF files and data compression using \*SAVF files, refer to the links below:

- [CRTSAVF](#)
- [SAVLIB](#)

DTACPR	Compression Type
*NO	None
*LOW	SNA
*MEDIUM	TERSE
*HIGH	LZ1
*ZLIB	ZLIB

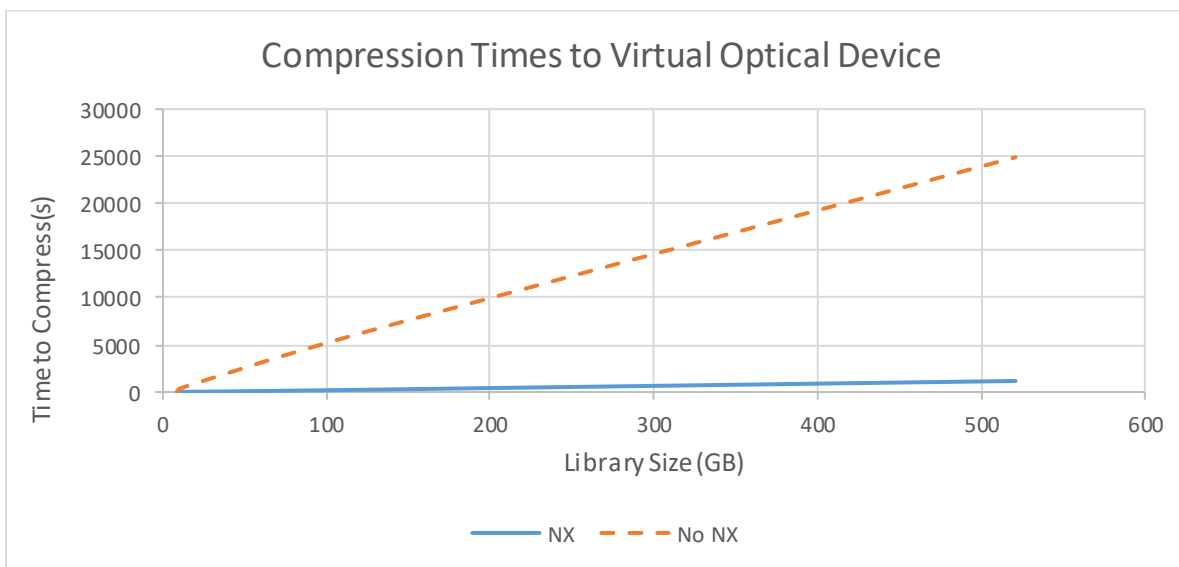
**Figure 24 - Compression Options**

Above is a list of the current options for the DTACPR argument on SAV operations, and the compression type each of the options maps to.

**B) How do I maximize data compression performance when saving to virtual optical media?**

If you’re looking to save backups of files or compress data for transfer, you may want to consider saving to Virtual Optical Media. Virtual Optical Media supports all compression types offered on IBM i (\*ZLIB, \*HIGH, \*MEDIUM, \*LOW, and \*NO). As mentioned in [Section 7.23](#), an IBM i partition running on a Power 10 system can use the NX accelerator for DEFLATE (\*ZLIB) compression. When saving to a virtual optical device on IBM i, SAVLIB DTACPR(\*ZLIB) can run up to 20 times faster on Power 10 than Power 9 (for both small and large libraries). If using Power 9, SAVLIB DTACPR(\*MEDIUM) will offer the best ratio between compressibility and time taken to complete compression. More information on setting up a virtual optical device and saving a library to a virtual optical device can be found here:

[Virtual Optical Media SAVLIB](#)



**Figure 25 - Compression Times by Library Size**

The chart above follows the save times for four libraries ranging from 8 GB to 500 GB. The top line outlines the time to save without NX support, while the bottom lines outline save times with NX. Notice how the difference can be up to 20x faster when running with NX.

**7.24 Should I utilize ASP Encryption?**

New hardware-assist function recently added to IBM i 7.5 and the latest Power systems (Power9 and Power10) provides improved performance for storage access operations when ASP encryption is enabled. Workloads that require frequent storage accesses (read/write/update operations) can experience significantly improved storage I/O response times and reduced CPU consumption from this new functional enhancement. In addition, the encrypted ASP storage I/O throughput rate may approach the same rate as unencrypted ASP storage I/O operations. The CPU usage may be reduced

by as much as half for large record I/O operations.

More information on how to create encrypted ASPs can be found here:

[Creating an encrypted auxiliary storage pool](#)

## 7.25 Where can I find more information on performance tuning?

The IBM Docs is a great resource for learning about performance tuning. The top-level link for tuning system performance is:

[Monitoring and Managing Performance](#)

The [References](#) chapter has many more links related to performance.

## 7.26 IBM Technology Expert Labs can help

The [IBM Technology Expert Labs](#) organization

<https://www.ibm.com/it-infrastructure/services/lab-services>

is available to assist customers with resolving system, application, and database performance problems. Formal and informal training opportunities are also available where customers learn how to use the performance tools and resolve performance problems on their own. The IBM i performance team can be contacted via email to Eric Barsness at [ericbar@us.ibm.com](mailto:ericbar@us.ibm.com).

The IBM i Db2 [for i](#) team in IBM Technology Expert Labs is also available to help clients in all areas of Db2 on i including Business Intelligence (BI), Analytics, ~~Db2 Web Query~~, database modernization, Very Large Database (VLDB) design, performance, and more. Contact Kent Milligan at [kmill@us.ibm.com](mailto:kmill@us.ibm.com).

## 8 Database Performance

### 8.1 Are there some basic suggestions for optimal SQL Performance?

The following suggestions apply to any SQL workload including embedded SQL in high-level language (HLL) programs, JDBC, ODBC, Db2 Web Query for i, etc.:

- Make sure you are using the latest level of Db2 for i. This ensures that most or all the SQL processing will use the newer SQL Query Engine, as well as many other performance enhancements delivered in recent IBM i OS releases and technology refreshes. Review recent performance enhancements in Db2 for i at:  
[Recent Performance Enhancements - Db2 for i Technology Updates](#)
- Implement an appropriate indexing strategy based on index advice from the query optimizer and the overall profile of the database. Index advice is available from Visual Explain, Db2 for i catalogs, the SQL plan cache, and the SQL performance monitor. Read the following White Paper:  
[IBM Db2 for i indexing methods and strategies](#)
- Utilize the [MTI\\_INFO](#) service to identify temporary indexes that have been created by Db2 for i. The information returned by this service enables the creation of permanent indexes to replace the temporary indexes which will reduce temporary storage usage and will provide more consistent query performance.
- Verify that the available “fair share of memory” is large enough to ensure efficient query implementation. This value is calculated by the optimizer based on the size of the memory pool, the number of active jobs, and sometimes by the configured activity level depending on the Db2 for i level. The number of active jobs calculation for a memory pool is more accurate when the memory pool is using Expert Cache.
- Ensure that the optimization goal is consistent with the intended use of the query (\*ALLIO, \*FIRSTIO).
- For environments dominated by short running queries, determine if large numbers of full opens and/or access plan rebuilds are impacting performance.
- For environments featuring a high-number of SQL procedural objects (functions, procedures, and triggers), review the object attributes and procedural coding to determine if more efficient options and coding practices can be utilized. Read the following White Paper: [Improving SQL Procedure Performance](#)
- Consider selectively enabling Db2 Symmetric Multiprocessing feature of IBM i (5770SS1 Product option 26) to allow for parallel processing on longer-running queries. When selectively enabling the use of parallel processing, it's recommended to use a parallel degree setting of \*OPTIMIZE. IBM i 7.5 provides additional function and tuning capabilities when using parallel processing.  
<https://www.ibm.com/support/pages/sqe-improved-smp-processing>

This product (5770SS1 Product option 26) is now available at no charge.

- The Query Supervisor feature can be used to limit the amount of time that a query can run or the amount of system resources a query can consume. This tool can also be used to just track the queries that are running long or consuming significant resources.  
<https://www.ibm.com/docs/en/i/7.5?topic=tools-query-supervisor>

The following apply to applications which use an explicit connection model such as JDBC, ODBC, and CLI:

- Review the prestart job entries for QZDASOINIT and QSQSRVR and make appropriate adjustments. Prestart job activity can be monitoring using the DSPACTPJ command or the [PRESTART\\_JOB\\_STATISTICS table function](#).
- If necessary, selected QZDASOINIT and QRWTSRVR jobs can be rerouted to different subsystems using the [SET\\_SERVER\\_SBS\\_ROUTING procedure](#).
- Ensure you are using connection pooling. Review the QHST log or run database monitor to determine if there is a problem. The [HISTORY\\_LOG\\_INFO table function](#) can be used to do this.

## 8.2 How can I improve performance of applications that use Record Level Access (Native I/O)?

See the chapter [RPG/COBOL and Native I/O](#)

## 8.3 What tools are available to monitor and tune SQL?

IBM i provides several integrated tools to collect and visualize database specific performance information. They can help you identify and tune performance problem areas. Key tools available include:

- [IBM i Access Client Solutions](#) (ACS) contains several components that a database engineer, administrator, or developer should leverage to address tuning of SQL.

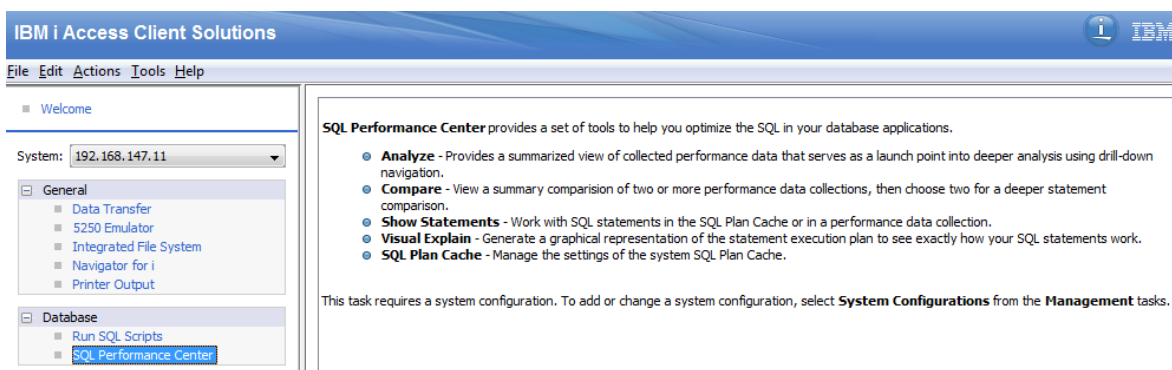


Figure 26 - SQL Performance Center

- **SQL Plan Cache** collects plans in real time for SQL statements executing queries including non-SQL query interfaces such as Open Query File (OPNQRYF). It is continuously collecting information and provides a “moving picture” of query activity. You can view the “live” plan cache or save point in time snapshots of the plan cache for later analysis or comparisons. Plan Cache analysis is available from the SQL Performance Center link in ACS.

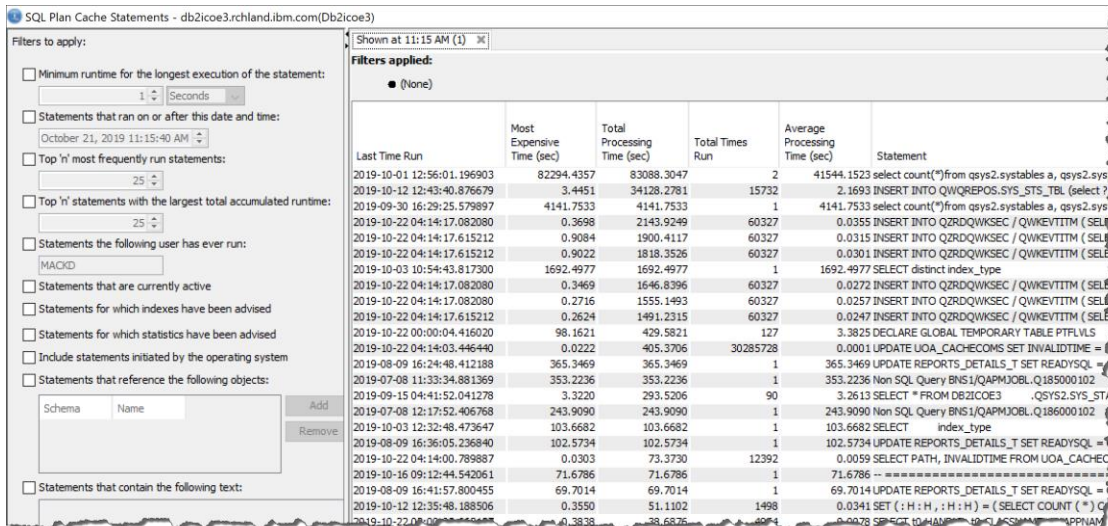


Figure 27 - SQL Plan Cache

The Show Statements option provides an ability to filter on the plan cache data and displays “plans” based on your filter criteria.

For additional information: [IBM Docs: SQL Plan Cache](#)

- **SQL Database Monitor** – This monitor is a collection process that must be started and then turned off. It gathers real-time information about all SQL statements and non-SQL queries. The collection information includes details about optimization and runtime behavior. Its contents are viewable through the SQL Performance Center in the IBM i Access Client Solutions.

For additional information, refer to [IBM Docs: Database Monitor](#)

SQL Performance Center - db2icoe3.rchland.ibm.com  
 File View Actions Tools Help  
 Database: Db2icoe3  
 Plan Cache Performance Monitors Plan Cache Snapshots Plan Cache Event Monitors

Name	Schema	Table	Created By	Status	Date Created
QIBMDB2COEQDBMNCQE011643682108	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE021643682111	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE031643682113	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE041643682114	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE051643682115	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE061643682118	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE071643682120	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE081643682121	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/09/2016
QIBMDB2COEQDBMNCQE091643682122	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/10/2016
QIBMDB2COEQDBMNCQE101643682125	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/10/2016
QIBMDB2COEQDBMNCQE111643682127	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/10/2016
QIBMDB2COEQDBMNCQE121643682128	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/10/2016
QIBMDB2COEQDBMNCQE131625052068	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE141625052071	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE151625052073	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE161625052074	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE171625052075	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE181625052078	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE191625052080	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE201625052081	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE211625052082	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016
QIBMDB2COEQDBMNCQE221625052083	QIBMDB2COE	QDBMNCQ...	MCKINLEY	Ended	03/05/2016

Figure 28 - SQL Database Monitor

- **Visual Explain** – Within Show Statements (either the SQL Plan Cache or a database monitor) you can select Visual Explain to analyze a query. Visual Explain provides a picture of how Db2 is processing a query request. A tremendous amount of information is provided here to diagnose a poor performing query.

For additional information: [IBM Docs: Visual Explain](#)



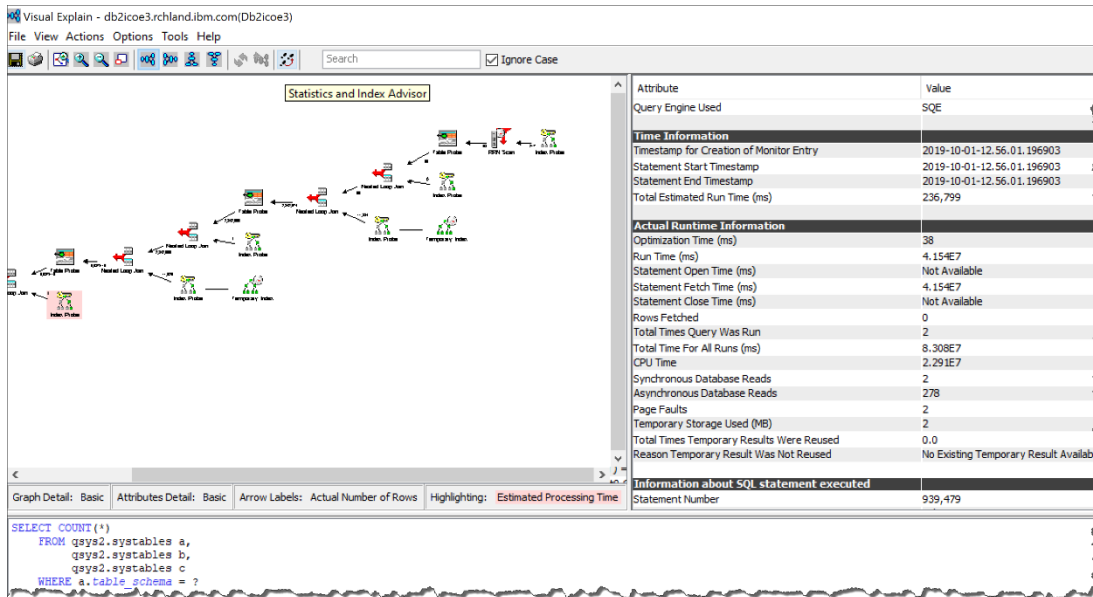


Figure 29- Visual Explain

- **IndexAdvisor (of a statement)** – Within Visual Explain of a statement (either the SQL Plan Cache or a database monitor collection) you can select Index Advisor. Indexing is a critical performance enhancement technique in Db2 for i, and Db2 provides recommendations for indexes that would help the selected query request.

For additional information: [IBM Docs: Index Advisor](#)

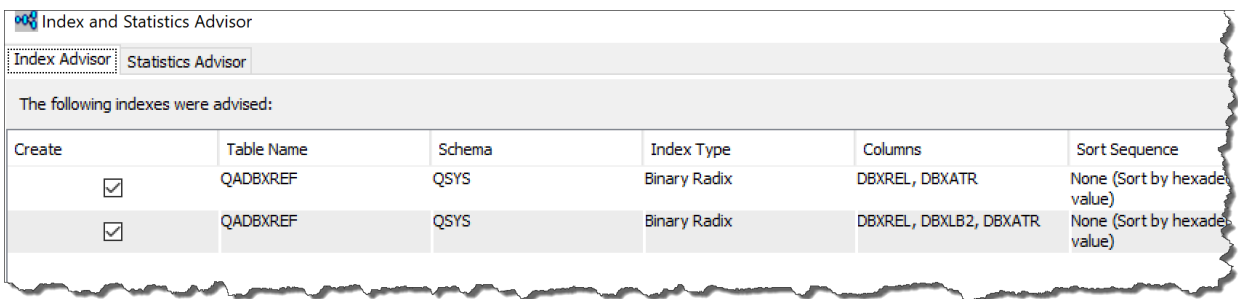


Figure 30 - Index Advisor

- **Index Evaluator** – With the ACS Schemas tool, you can evaluate what indexes are providing value to the query optimizer and query engine. The Index Evaluator provides usage information on all of the indexes that are available for the optimizer to use. The Index Evaluator is launched by right-clicking a table object and selecting the **Work With -> Indexes** task to produce the following index usage output.

Name	Schema	Type	Last Query Use	Last Query Statistics Use	Query Use Count	Query Statistics Use Count	Last Used
IDX_EVLT_PLANT_NAME	QWQCENT	Index	10/29/2020 08:36:05 AM	12/17/2020 02:55:58 PM	23	267	10/29/2020
IDX_PLANT_INFO	QWQCENT	Index	12/05/2018 04:06:04 PM	02/24/2021 12:25:40 PM	5	127	12/05/2018
Q_QWQCENT_PLANT_PLANT_CODE_00001	QWQCENT	Primary Key Constraint	02/22/2021 04:21:03 PM	02/24/2021 03:38:23 PM	544	14,158	02/24/2021

Figure 31 - Index Evaluator

- Collection Services** – Many database and SQL related metrics are collected by Collection Services (Collection Services should be automatically running on your partition 24x7). Several job level SQL metrics are captured. Metrics include:
  - SQL clock time
  - SQL unscaled / scaled CPU time
  - SQL synchronous / asynchronous database reads
  - SQL synchronous / asynchronous non-database reads
  - SQL synchronous / asynchronous database writes
  - SQL synchronous / asynchronous non-database writes
  - Number of high-level SQL statements per thread

These metrics provide information regarding how much resource usage (such as CPU and I/O) was due to SQL work being performed by the thread. This data can be visualized in Performance Data Investigator (PDI). Launch to Investigate Data and then filter by the Database Package:

The screenshot shows the PDI interface with the 'Package Name' filter set to 'Database'. The 'Perspective' dropdown is set to 'I/O Reads and Writes'. The 'SQL' column displays various SQL queries related to database performance metrics. The 'Database' package is selected in the left-hand navigation pane, and the main area displays a list of perspectives for that package, including 'I/O Reads and Writes', 'SQL CPU Utilization Overview', 'Database Locks Overview', and several 'Physical Database I/O' views. A tooltip is visible over the 'Physical Database I/O by Subsystem - Basic' perspective, showing a navigation path: Database --> Database I/O --> Physical Database I/O by Subsystem - Basic.

Figure 32 - PDI Database Package

The following is an example chart of Detailed Physical Database I/O by Job or task showing the amount of reads/writes per second that are SQL related versus non-SQL related activities:

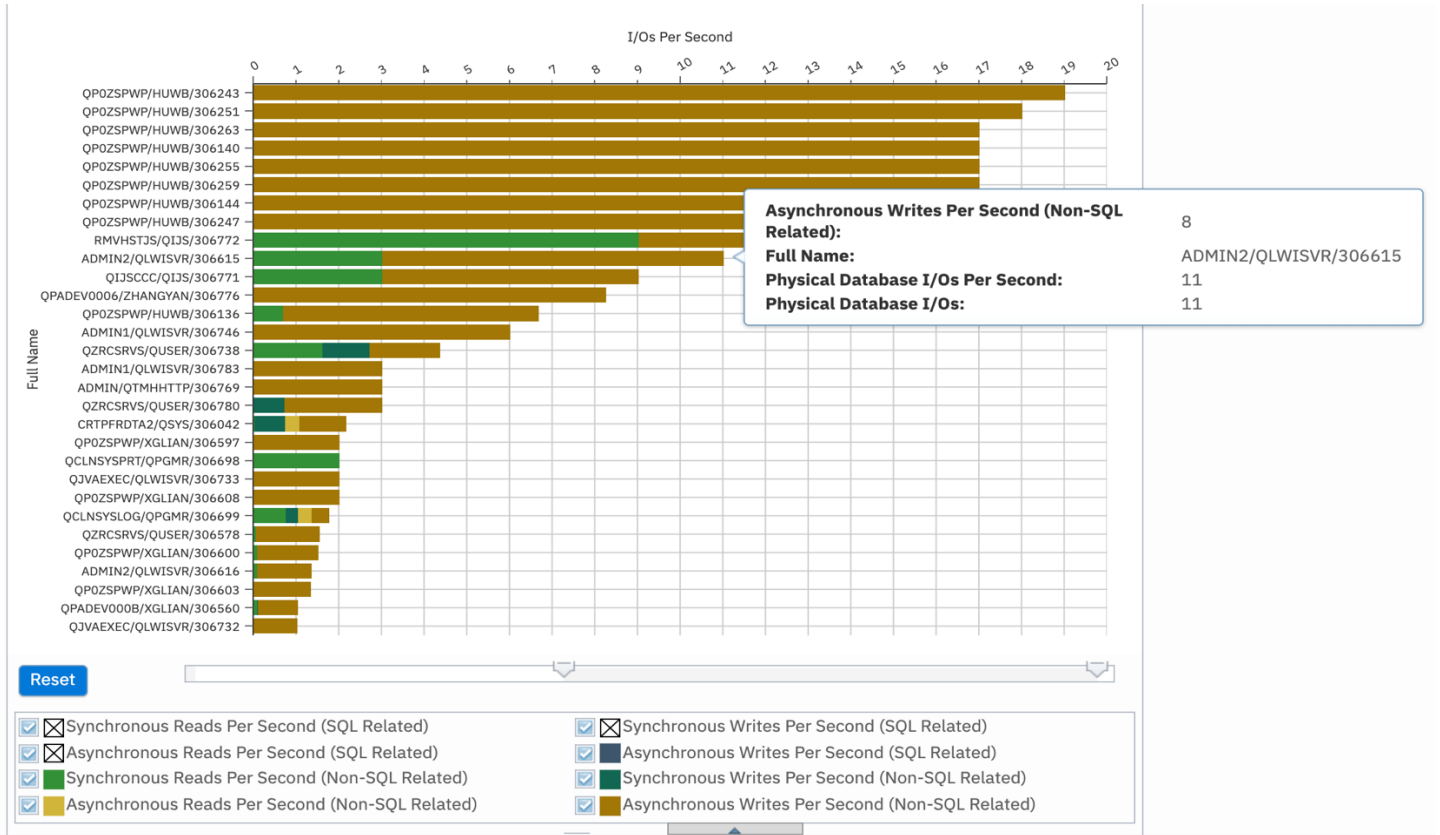


Figure 33 - PDI Physical Database I/O by Job or Task

Starting with IBM i 7.2, Collection Services has included a new SQL category; this SQL category includes SQL Plan Cache data. This category is part of the standard collection category, and this data is collected by default. This new data can be viewed in PDI in the Database package as shown here:

Package Name	Perspective	Description
Database	Filter	collection or collections
Database	<a href="#">SQL Cursor and Native DB Opens Overview</a>	This chart shows the number of SQL cursor opens, SQL pseudo opens, Native database (non-SQL) full opens as rates per second that occurred over time for the selected collection or collections. Use this chart to select a time frame for further detailed investigation.
Database	<a href="#">Query Opens</a>	This chart shows the number of full and pseudo query opens that occurred over time for the selected collection or collections. The number of queries that were hard closed during this time is also shown.
Database	<a href="#">Active Query</a>	This chart shows the total number of active queries over time for the selected collection or collections.
Database	<a href="#">Plan Cache Searches</a>	This chart shows the total number of plan cache searches done over time for the selected collection or collections. The total number of plan cache searches is broken down into searches resulting in plans found and plans not found. The number of query full opens and queries hard closed is also shown.
Database	<a href="#">Plans Detailed</a>	This chart shows the number of plans stored in the SQL plan cache, plans built, plans pruned due to plan cache size, and plans removed from SQL plan cache over time for the selected collection or collections. This chart also shows the size of the SQL plan cache and threshold value for SQL plan cache. Plan cache size threshold is the maximum size that the SQL plan cache is allowed to be before DB2 automatically manages the SQL plan cache and replaces older plans with new plans.
Database	Database --> SQL Performance Data --> Collection Services --> Maintained Temporary Indexes (MTIs)	
Database	<a href="#">Maintained Temporary Indexes (MTIs)</a>	This chart shows the number of Maintained Temporary Indexes (MTIs) created and deleted over time for the selected collection or collections.
Database	<a href="#">Adaptive Query Processing (AQP)</a>	This chart shows the number of executing queries checked by Adaptive Query Processing (AQP) and the number of plans replaced by AQP over time for the selected collection or collections. The number of queries that were opened during this time is also shown.

Filtered Rows: 7 | Total Rows: 475

Figure 34 - PDI - Collection Services perspectives in Database Package

The following is an example Query Opens chart showing number of Opens being done across the Collection Services time intervals:

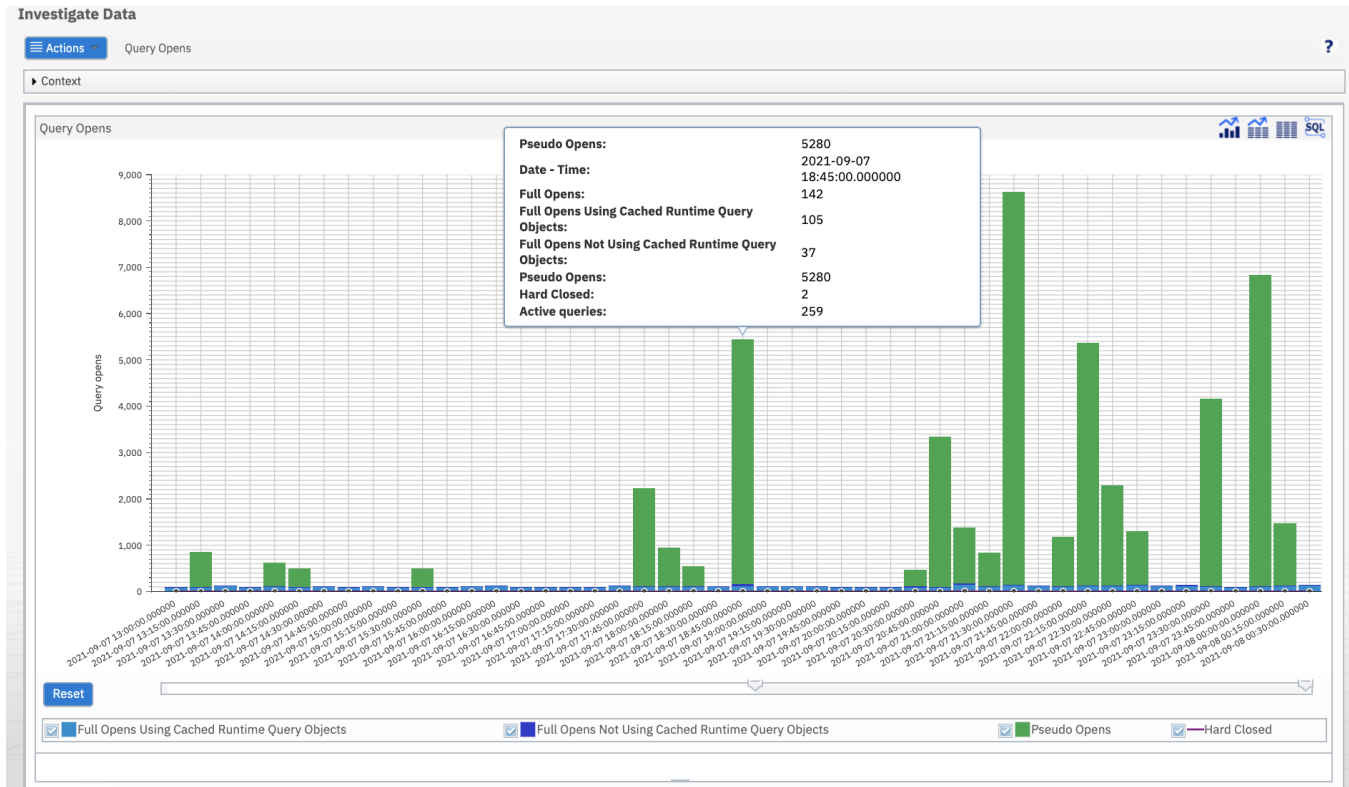


Figure 35 - Collection Services - Query Opens Chart

There are many other Database and SQL charts available. See the following for more information:

[Database Package and Perspectives](#)

## 8.4 IBM Db2 Mirror for i

Db2 Mirror is a continuous availability solution available in IBM i 7.4 and IBM i 7.5. More information about Db2 Mirror can be found here:

[IBM Docs - Db2 Mirror.](#)

[IBM Db2 Mirror for i: Performance Considerations](#)

Db2 Mirror requires a high-speed network connection to perform synchronous data operations. The high-speed connection interface being used is Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCE). Network Redundancy Groups (NRGs) are used to automate the physical network redundancy, failover, and load balancing of the high-speed connections.

The new Collection Services collection category \*RDMA collects performance information about NRGs and RDMA links. The \*RDMA category is collected when Collection Services is configured to collect the

\*STANDARD collection profile. The new Collection Services data files are:

- [QAPMLINK](#)
- [QAPMNRG](#)
- [QAPMNRGL](#)

## 8.5 More information on Db2 for i performance and support

The Db2 for i overview information can be found at:

[IBM Support - Db2 for i](#)

and

[IBM Docs - Db2 for i](#)

The IBM Docs has a section on SQL performance:

[SQL Performance](#)

There are more links related to database performance in the [References](#) chapter.

IBM Technology Expert Labs Db2 for i consulting team can also provide guidance around performance-related issues or information transfer. A list of Db2 for i IBM Technology Expert Labs offerings can be found [here](#).

## 9 IBM i Services

### 9.1 Introduction

There are many system services that can be accessed through system-provided SQL views, procedures, and functions. These provide an SQL interface to access, transform, order, and subset the information without needing to code to a system API.

In order to take advantage of the IBM i Services, you need a basic understanding of SQL. The [Db2 for i SQL Reference](#) is a good place for getting started with SQL as well as a great reference for those who are experts.

Certainly, some of the IBM i Services just provide another way to get at information you can already access via commands or APIs. But there are some services that provide access to information that otherwise might be difficult to get. A few examples are:

- [System Health Services](#) provide the capability to programmatically get at system limit information. Are you near the system limits for files or jobs or journals (or many other limits)? How do you know? System health services can help you answer that question.
- System temporary storage ([SYSTMPSTG](#)) service provides a programmatic interface to identify where the temporary storage allocations have been made.
- [PTF Group Currency](#) service provides a capability to check your installed PTF groups with the levels that are available on Fix Central

### 9.2 IBM i services currently available

- [Application Services](#)  
These procedures functions, and views provide interface information that can be used by applications.
- [Backup, Recovery, and Media Services \(BRMS\) Services](#)  
These table functions and views provide information about BRMS.
- [Communication Services](#)  
These services provide communication information.
- [IFS Services](#)  
These services provide information about the integrated file system (IFS).
- [Java Services](#)  
This view and procedure provide Java information and JVM management options.
- [Journal Services](#)  
This function and view provide journal information.
- [Librarian Services](#)  
These services provide object and library list information.
- [Message Handling Services](#)  
These views and functions provide system message information.
- [Performance Services](#)

These services provide information related to system performance.

- [PowerHA Services](#)  
These table functions and views provide information about PowerHA®
- [Product Services](#)  
These services provide information about licensed products.
- [PTF Services](#)  
These views provide PTF information.
- [Security Services](#)  
These views, procedures, and functions provide security information.
- [Spool Services](#)  
This view and function provide information about spooled files.
- [Storage Services](#)  
These views provide information about storage and storage devices.
- [System Health Services](#)  
For the most important system resources, the IBM i operating system automatically tracks the highest consumption and consumers.
- [Work Management Services](#)  
These views and functions provide system value and job information.

### 9.3 Db2 for i Services currently available

- [Application Services](#)  
These procedures provide interfaces that are useful for application development.
- [Performance Services](#)  
These services include procedures that provide interfaces to work with indexes and a view to see information about database monitors. In addition, the QSYS2.OVERRIDE\_QAQQINI procedure can be used to establish a temporary version of the QAQQINI file.
- [Plan Cache Services](#)  
These services include procedures to assist you in performing database administration (DBA) and database engineering (DBE) tasks.
- [Utility Services](#)  
These procedures provide interfaces to monitor and work with SQL in jobs on the current system or to compare constraint and routine information across systems.

### 9.4 IBM i Access Client Solutions

IBM i Access Client Solutions is the newest member of the IBM i Access family. It provides a Java based, platform-independent interface that runs on most operating systems that support Java, including Linux, Mac, and Windows™.

IBM i Access Client Solutions consolidates the most commonly used tasks for managing your IBM i into one simplified location. The latest version of IBM i Access Client Solutions is available to customers with an IBM i software maintenance contract.

[Product Download Information](#)



This tool provides an interface to access the IBM i Solution functions. It also includes examples to show how to build SQL statements to retrieve results.

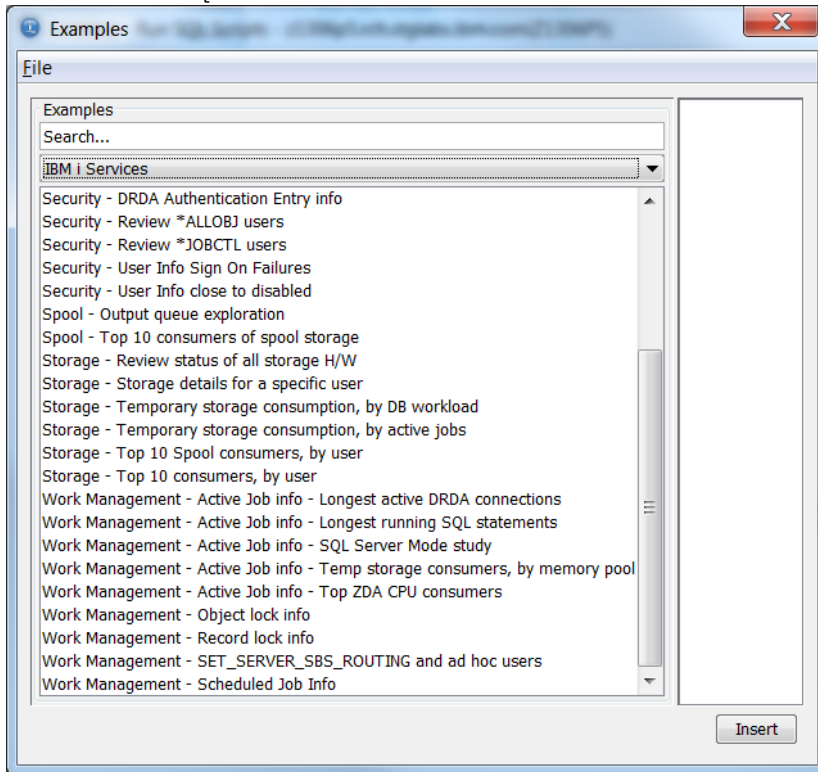


Figure 36 - IBM i Access Client Solutions - IBM i Services Examples

Following is the SQL window showing how to use the IBM i Access Client Solutions tool to find the longest running SQL statements using one of the examples above:

```

1  -- category: IBM i Services
2  -- description: Work Management - Active Job info - Longest running SQL statements
3  --
4  -- Find the jobs with SQL statements executing and order the results by duration of SQL statement execution
5  --
6  WITH ACTIVE_USER_JOBS (Q_JOB_NAME, AUTHORIZATION_NAME, CPU_TIME, RUN_PRIORITY) AS (
7      SELECT JOB_NAME,
8             AUTHORIZATION_NAME,
9             CPU_TIME,
10            RUN_PRIORITY
11          FROM TABLE (
12             ACTIVE_JOB_INFO('NO', '', '', '')
13          ) x
14         WHERE JOB_TYPE <> 'SYS'
15      )
16  SELECT Q_JOB_NAME,
17         AUTHORIZATION_NAME,
18         CPU_TIME,
19         RUN_PRIORITY,
20         V_SQL_STATEMENT_TEXT,
21         ABS(CURRENT_TIMESTAMP - V_SQL_STMT_START_TIMESTAMP) AS SQL_STMT_DURATION,
22         B.*
23  FROM ACTIVE_USER_JOBS.

```

Q_JOB_NAME	AUTHORIZATION_NAME	CPU_TIME	RUN_PRIORITY	V_SQL_STATEMENT_TEXT
034400/QUSER/QZDASOINIT	PDITEST	851	20	SELECT COUNT(*) FROM (SELECT MAX(JBNAME) AS JBNAME, MAX(JBUSER) AS JBUSER, MAX(JBNBR) AS JBNBR, MAX(JBT..
034361/QUSER/QZDASOINIT	STACYB	796	20	-- category: IBM i Services-- description: Work Management - Active Job info - Longest running SQL ..

Figure 37 - IBM i Access Client Solutions – Active Job Info – Longest running SQL statements

## 9.5 For more information:

Here's the launch point for IBM i Services and Db2 for i Services.

[IBM i Services](#)

[Db2 for i Services](#)

On the IBM Support website, this shows the details on the enabling operating system level and Db2 PTF Group: <http://ibm.biz/Db2foriServices>

## 10 RPG/COBOL and Native I/O

### 10.1 Introduction

This section discusses methods for improving the performance of applications written in RPG or COBOL and those programs that are accessing data using native IO operations.

### 10.2 CPU performance tips

RPG and COBOL programs are optimized to run on IBM i with every release and will adopt the latest processor enhancements when compiled or when re-translated as required, for example when moving from IBM i 5.4 to 6.1 or 7.1.

However, additional optimization can be done over and above what the compilers and the translator provide in order to improve performance in terms of runtime or CPU. Some of the following optimization techniques will trade off better optimization with the capability to debug the code, so be aware when deciding to use program optimization. It is generally good practice to select a few highly used programs with intense use of CPU cycles and perform optimization and when debugging is not required any longer. Some of these performance optimization techniques can be very significant and worth the effort for highly used programs in the interactive or batch environment.

#### 10.2.1 *Compiler Options*

Optimize the use of processors by specifying the OPTIMIZE parameter on the CRTRPGMOD and CRTBNDRPG command, e.g. OPTIMIZE(\*BASIC or \*FULL). You can also use the commands CHGMOD, CHGPGM or CHGSRVPGM and the parameter OPTIMIZE(\*YES) to optimize instructions being executed within your program. Note that debug mode function is reduced as optimization is added. For more information, see:

[Create Bound RPG Program](#)

Optimize the use of program arguments by using the parameter ARGOPT(\*YES) on the CRTPGM and CRTSRVPGM commands. For more information, see:

[Improve Performance with Argument Optimization](#)

Control program activation by the activation keyword (\*DEFER) with the bind service program parameter on the CRTPGM, CRTSRVPGM, UPDPGM and UPDSRVPGM commands. For more information, see:

[Activate Your Programs On Demand](#)

#### 10.2.2 *Profiling*

Optimize your ILE RPG or COBOL programs by profiling and collecting procedure execution information and applying them to optimize the program structures and instruction flows. For more information read:

[Program Profiling](#)

Identifying hot programs (programs which use most of the resources) within your application with Performance Explorer and the PEX definition \*STATS. For more information, refer to the following IBM Redbooks:

[Application and Program Performance Analysis using PEX Statistics on IBM i5/OS](#)

Identifying hot statements of programs (actual statements of programs which use most of the resources) within your application with Performance Explorer and the PEX definition \*PROFILE. For more information, refer to the following IBM Redbooks:

[Application and Program Performance Analysis Using PEX Statistics on IBM i5/OS](#)

## 10.3 IO performance tips

### 10.3.1 *General tips*

Use higher blocking for input/output only operations – SEQONLY((\*YES #rcds)) >= 128KB. More information on increasing logical page size in DB operations can be found here:

[Sequential-only processing of database files](#)

Avoid open/close of files, use RETRN vs SETON \*LR in RPG

[Tips: Run and call IBM i procedures](#)

Eliminate FRCRATIO(1) to prevent every update/write to be forced out to disk

[FRCRATIO parameter](#)

The number of physical I/Os can be significantly reduced by using the SETOBJACC command and pinning a database file or index into a private memory pool which is not used by anything else. Read more at:

[SETOBJACC/CLRPOOL Commands](#)

### 10.3.2 *Database modernization*

For applications which are dominated by database processing, performance can often be improved by converting existing high-level language programs from native IO (record level access) to SQL set-based processing. [IBM Technology Expert Labs](#) has a well-established process for this kind of database modernization which takes advantage of the SQL query optimizer's advanced capabilities while minimizing the impacts to existing programs.

## 11 C Program Compiler Optimization

C and C++ programs are optimized to run on IBM i with every release and will adopt the latest processor enhancements when compiled or when re-translated, e.g. when going from 5.4 to 6.1 to 7.1. Additional optimization can be done over and above what the compilers and the translator provide in order to improve performance in terms of runtime or CPU use and path length.

Be aware that some of the following optimization techniques trade off improved performance with less capability to debug the code. It is generally a good practice to focus optimization on a few highly used programs with intense use of CPU cycles when debugging is not required any longer. Some of these performance optimization techniques can be very significant and worth the effort for highly used programs in the interactive or batch environment.

### 11.1 IBM i Architecture – MI Interface

The IBM i Architecture depicted below only allows predefined MI Instructions to go to the lower level of microcode. All user code, including the compilers as well as the operating system itself, is independent from the microcode and hardware layer below. All compiled code will have to get translated in order to get machine level instructions specific to the hardware being run on.

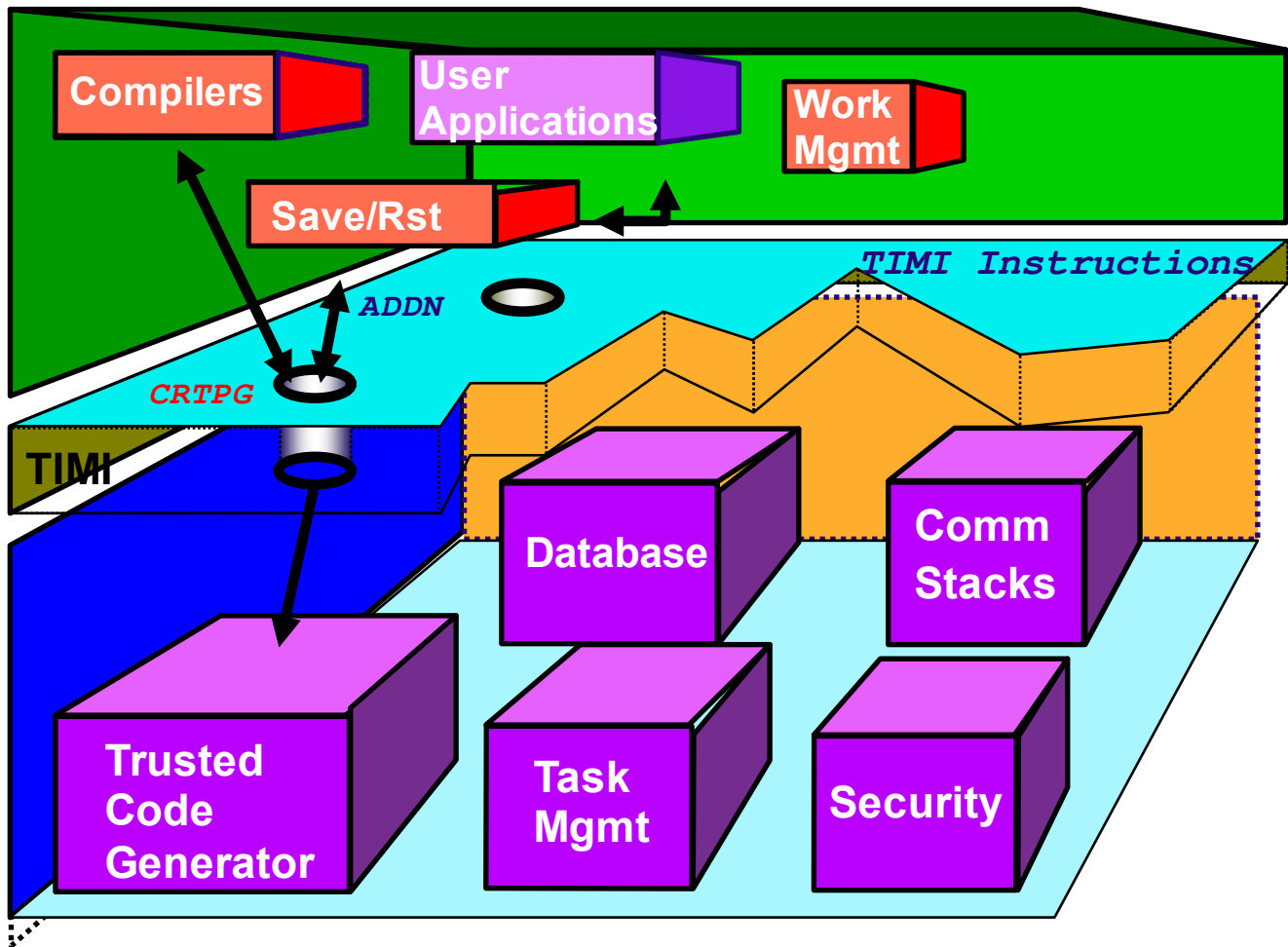
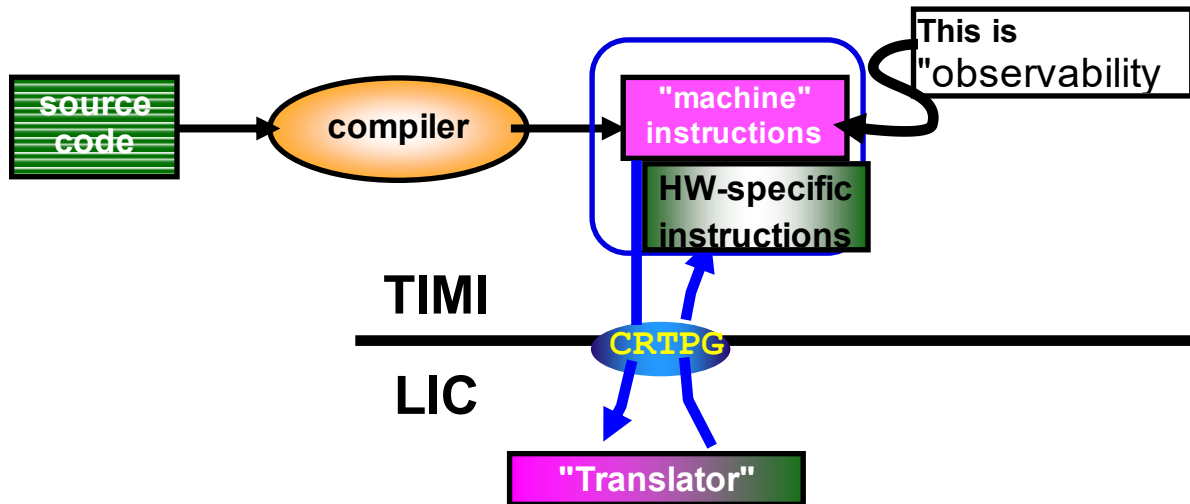


Figure 38 – IBM i Architecture – MI Interface

## 11.2 IBM i Program Creation

During the compile of any program, including C and C++ programs, you will need to use the CRTPG MI instruction in order to generate machine executable code. This translation step can also be independently performed via the CHGPGM, CHGMOD, etc. commands.



**Figure 36 - IBM i Program Creation**

Key notes:

- Programs get compiled to an intermediate level of code - MI
- The IBM i architecture incorporates "translation" under the TIMI
- The Translator is necessarily hardware-specific

## 11.3 Compiler option - OPTIMIZE

Optimize the use of CPU by specifying the OPTIMIZE parameter on the CRTCMOD and CRTBNDC command, e.g. OPTIMIZE(20, 30 and 40).

Optimization may rearrange or eliminate some statements altogether. Consequently, field values presented by the debugger, as well as breakpoints and step locations, may be inaccurate.

### CRTCMOD and CRTBNDC commands

**OPTIMIZE(10)** is the [default value](#). The compiler performs no optimization on the code. This option allows you to debug the program accurately, with the debugger presenting correct current field values.

**OPTIMIZE(20)** performs [some optimization](#) on the compiled code. The debugger will let you display variables, but the displayed value may not be the current value. The debugger will also allow variable changes, but you may encounter unexpected results if you try to use the changed variables.

**OPTIMIZE(30)** generates more efficient code than level 20. More optimizations are performed in

addition to those performed at level 20, and the compile process takes longer. The debugger will let you see (but not change) field values, but the debugger may not show you the correct current field values.

**OPTIMIZE(40)** generates the most efficient code. All optimizations done at level 30 are performed on the generated code. In addition, code is eliminated from procedure prologue and epilogue routines that enable instruction trace and call trace system functions. Eliminating this code enables the creation of leaf procedures. A leaf procedure is a procedure that does not require an invocation frame to be stacked. Procedure call performance to a leaf procedure is significantly faster than to a normal procedure.

### **CHGMOD, CHGPGM or CHGSRVPGM commands**

You can also use the commands CHGMOD, CHGPGM or CHGSRVPGM and the parameter OPTIMIZE(\*YES) to optimize instructions being executed within your program. The DSPMOD, DSPPGM, and DSPSRVPGM commands shows you the current level of optimization for an object.

You can change optimization levels with the CHGMOD, CHGPGM, or CHGSRVPGM commands; these commands support the levels already discussed, but also add other options:

**OPTIMIZE(\*NONE|\*NO|10)** -- These options are equivalent, and are discussed above. OPTIMIZE(\*NO) is used only by the CHGPGM command.

**OPTIMIZE(\*BASIC|20)** -- These options are equivalent, and are discussed above.

**OPTIMIZE(\*FULL|30)** -- These options are equivalent, and are discussed above.

**OPTIMIZE(\*YES|40)** -- This is the highest possible optimization level. In addition to all level 30 optimization, level 40 performs optimization that reduces the capability to provide extensive call and instruction tracing.

**OPTIMIZE(\*YES)** is used only by the CHGPGM command.

Note that debug mode function is reduced as optimization is added.

More Information how to use the CRTBNDC command:

[Create Bound C Program \(CRTBNDC\)](#)

## **11.4 Program profiling**

Optimize your ILE C programs by profiling and collecting procedure execution information and applying that information to optimize the program structures and instruction flows. This is an advanced optimization technique that reorders procedures, or code within procedures, in ILE programs and service programs based on statistical data gathered while running the program. This reordering can improve instruction cache utilization and reduce paging required by the program, thereby improving performance.

Note: The semantic behavior of the program is not affected by program profiling.

The performance improvement realized by program profiling depends on the type of application and you can expect more improvement from programs that spend the majority of time in the application code itself, rather than spending time in the operating system or doing input/output processing.

The performance of program code produced when applying profile data depends upon the optimizing translator correctly identifying the most important portions of the program during typical use. Therefore, it is important to gather profile data while performing the tasks that will be performed by end users and using input data that is similar to that expected in the environment in which the program will be run.

Note: Program profiling is available only for ILE programs and service programs that are compiled using optimization level 30 or 40. Because of the optimization requirements, you should fully debug your programs before using program profiling.

To enable the program to collect profiling data:

- Compile your ILE program(s) using the PRFDTA(\*COL) parameter option.
- Start the program profiling collection on the system with the Start Program Profiling (STRPGMPRF) command.
- Collect profiling data by running the program through its high-use code paths. Because program profiling uses statistical data gathered while running the program to perform these optimizations, it is critical that this data is collected over typical uses of your application.
- End the program profiling collection on the system with the End Program Profiling (ENDPGMPRF) command.
- Apply the collected profiling data to the program by requesting that code be reordered for optimal performance based on the collected profiling data using the PRFDTA(\*APYALL) parameter option.

More Information how to find program hot spots can be found here:

[Program Profiling](#)

### 11.5 Advanced Argument Optimization

Advanced argument optimization is a cross-module optimization that is used to improve the performance of programs that contain frequently run procedure calls, including C++ applications that make mostly non-virtual method calls. Improved runtime performance is achieved by enabling the translator and binder to use the most efficient mechanisms to pass parameters and return results between procedures that are called within a program or service program.

The Argument optimization (ARGOPT) parameter, with \*YES and \*NO as possible values, is available on the CRTPGM and CRTSRVPGM commands to support advanced argument optimization. Specifying ARGOPT(\*YES) causes the program or service program to be created with advanced argument optimization. The default is \*NO.

Note: For programs that consist of hundreds or thousands of modules, creation time can be significantly longer using this option.

More Information how to perform argument optimization, read the following article:

[Improve Performance with Argument Optimization](#)



## 11.6 Interprocedural analysis (IPA)

At compile time, the optimizing translator performs both intraprocedural and interprocedural analysis.

**Intra**procedural analysis is a mechanism for performing optimization for each function within a compilation unit, using only the information available for that function and compilation unit.

**Inter**procedural analysis is a mechanism for performing optimization across function boundaries. The optimizing translator performs interprocedural analysis, but only within a compilation unit.

Interprocedural analysis that is performed by the IPA compiler option improves on the limited analysis described above. When you run interprocedural analysis through the IPA option, IPA performs optimizations across the entire program. It also performs optimizations not otherwise available at compile time with the optimizing translator.

In order to optimize your program or service program with the Inter-Procudural Analysis function, make sure that:

- All modules have the parameter MODCRTOPT(\*KEEPILDATA) specified and are compiled with an optimization level of 20 or higher
- And the parameter IPA(\*YES) is specified on the CRTPGM or CRTSRVPGM command

More information on additional optimization with IPA can be found in the IBM Docs at:

[Interprocedural analysis \(IPA\)](#)

## 12 Java and WebSphere

### 12.1 Introduction

This chapter covers best practices for Java and WebSphere Application Server performance. Most of the Java questions apply to WebSphere applications.

### 12.2 Does it matter if I use the 32-bit versus 64-bit JVM?

With current supported JVMs, there is little difference in performance between using 32 and 64-bit. When using a 64-bit JVM, compressed references (-Xcompressedrefs) can be used to reduce memory requirements. Newer versions of Java are 64 bit-only.

### 12.3 I'm using an old JDK level and am behind on PTFs, is that okay?

No, for best performance use the most current JDK level you can and should apply the latest PTFs.

### 12.4 Should I run my JVMs in a separate memory pool?

Yes, JVMs should run in a memory pool configured with enough memory to hold all JVMs at their maximum heap size. WRKSHRPOOL can be used to set the minimum pool size in percent to ensure the pool does not shrink too much if QPFRADJ is on. If there is not enough memory in the pool, the JVMs may become unresponsive due to high page faulting.

### 12.5 How do I determine the proper GC settings (heap sizes, collection policy, etc.) for my application?

For most customers, the GENCON policy results in the best performance. Discussion on how to set the minimum and maximum size is in the [IBM SDK for Java: Diagnostics Guide](#), section "How to do heap sizing".

Collection Services collects JVM information by default. Use Performance Data Investigator or iDoctor to visualize Java heap usage system-wide and by job. This might be a good starting point to find the job(s) which are consuming the most Java heap.

Take care not to overcommit the number of garbage collection (GC) threads for your JVM. The default number of GC threads depends on the number of cores in the partition. It is calculated by the number of dedicated processors, or virtual processors (VPs) for shared processor partitions, multiplied by the maximum SMT level minus 1. So a Power7 partition (with SMT level of 4) and with 20 VPs would have 79 helper threads per JVM by default. Each of these threads is concurrently active during garbage collection, which can lead to CPU queuing, and can limit the amount of CPU available for other work on the partition.

You can control the number of GC helper threads via the -Xgcthreads## setting. This sets the number of helper threads you want to be used during garbage collection. If you want to use 8 threads, then use -Xgcthreads08, which will result in one controlling thread and 7 GC helper threads.

Keep in mind that IBM i and the JVM determine the values for Java system properties by using the following order of precedence:

1. Command line or JNI invocation API
2. QIBM\_JAVA\_PROPERTIES\_FILE environment variable
3. user.home SystemDefault.properties file
4. /QIBM/UserData/Java400/SystemDefault.properties
5. Default system property values

Read more on the IBM Docs page: [Java system properties](#)

### **12.6 How do I analyze JVM dumps and verbose GC output?**

Download and install the IBM Support Assistant and install the IBM Monitoring and Diagnostic Tools for Java toolset: [Monitoring and Post Mortem/](#)

### **12.7 How do I tune Java code?**

The Java and WebSphere Performance on IBM eServer iSeries Servers Redbook still is a good resource even though it is from 2002. The recommendations specific to the JVM mostly do not apply anymore, but the coding tips and techniques are still valid.

<http://www.redbooks.ibm.com/abstracts/sg246256.html?Open>

### **12.8 Where can I get more information on Java tuning and debugging?**

The IBM support page collections have a wealth of information.

<https://www.ibm.com/support/pages/java-sdk-documentation>

### **12.9 Can I use Job Watcher to collect Java information?**

By default, Job Watcher does not collect the call stacks of Java threads. The “IBM Technology for Java” data collection tab in iDoctor can be used to enable the collection of JVM statistics and Java thread call stacks. IBM supplied definitions ending in “J” include the collection of this additional information.

### **12.10 How can I improve the performance of WebSphere Application Server running on IBM i?**

- Use the most recent version of WebSphere available.
- Turn off JSP reloading in production environments.
- Use the Fast Response Cache Accelerator (FRCA) for HTTP Server for caching static and dynamic content. Use the Apache HTTP server cache for secure (SSL) static content.

[Fast Response Cache Accelerator \(FRCA\) for HTTP Server](#)

The IBM Docs WebSphere Application Server contains information on tuning WebSphere:

[Tuning the application serving environment](#)

### **12.11 How do I tune IBM Toolbox for Java performance?**

See the IBM Docs section on this topic: [IBM Toolbox for Java Performance](#)

## 12.12 Where can I find additional information?

In the IBM Docs Java Performance section: [Java Performance for IBM i](#)

## 13 Power Processor-based Systems

### 13.1 Introduction

This chapter covers best practices for Power processor-based system performance.

### 13.2 Simultaneous Multithreading (SMT) and Intelligent Threads

Power8, Power9, and Power10 processors support Simultaneous Multithreading (SMT) with eight hardware threads, which provides more transaction processing capacity per core. Power hardware has had [SMT support](#) for many years. Initially, it was two hardware threads of execution on a processor core. With Power7, the number of hardware threads was increased to four. With Power8, the number of hardware threads was increased to eight.

#### 13.2.1 *Single thread (ST) versus SMT2/SMT4/SMT8*

Although it is not generally recommended for most customers, applications that are single process and single threaded may gain some benefit from running in ST mode. Multithreaded and/or multi process applications typically benefit more running in higher SMT modes. ST mode may also be beneficial in the case of applications where the number of application threads is smaller than the number of cores assigned to the partition. Applications that do not scale with a larger number of CPUs may also benefit from running in SMT2 or ST mode instead of SMT4 or SMT8 since lower number of SMT threads mean lower number of logical CPUs. This is a partition-wide setting and overall transaction processing capacity may be reduced by changing from the default mode.

The default mode on Power7 systems is SMT4. The default mode on Power8 systems running IBM i 7.1 TR8 is SMT4 and Power8 systems running IBM i 7.2 is SMT8. The default mode on Power9 and Power10 systems is SMT8.

For detailed information refer to:

[Under the Hood: Simultaneous Multi-Threading on POWER7 Processors](#)

#### 13.2.2 *Intelligent Threads Technology*

The Power10 processor, like Power9, Power8 and Power7 processors before it, uses Intelligent Threads technology to maximize workload performance regardless of the processor's threading context. This technology has been enhanced to adapt more quickly and with greater efficiency to changes in the workload. If the Power10 processor is under-committed, meaning fewer hardware threads are dispatched than are available, the core performance is roughly the same independent of threading context. So, for example, if one thread is dispatched, performance will be similar in ST, SMT2, SMT4, and SMT8 contexts; if two threads are dispatched, performance will be similar in SMT2, SMT4, and SMT8 contexts; if three or four threads are dispatched, performance will be similar in SMT4 and SMT8 contexts.

From a usability standpoint, Intelligent Threads means that system-level processor threading context adjustments typically aren't necessary in order to maximize workload performance. Highly multi-threaded workloads can benefit from the additional throughput offered by SMT8 technology, but multi-threading isn't necessary for a workload to achieve high-performance. It just works!

### 13.3 Flexible SMT Controls

SMT settings can be adjusted specifically for each partition. While the default processor threading context is suitable for most commercial environments, IBM i offers manual controls that allow the system to be fine-tuned to the specific characteristics of the workload. In general, as the processor threading context is decreased, single-thread performance may be improved at the expense of maximum aggregate throughput.

IBM i provides flexible, dynamic system-level processor threading controls. The processor threading contexts available for the partition IPL is determined by the Processor Compatibility Mode (PCM) partition attribute as shown in Table 1. Note that the PCM is established during partition activation.

#### 13.3.1 Processor Compatibility Mode

Power10, Power9, Power8 and Power7 systems can set the Processor Compatibility Mode to change the SMT behavior. Partitions running in Power6 or Power6+ compatibility mode can run in ST or SMT2. Partitions running in Power7 compatibility mode can run in SMT4, SMT2 or ST mode.

Processor Compatibility Mode	Supported Threading Contexts	Default Thread Context		
		7.3	7.4	7.5
Power7	ST, SMT2, SMT4	SMT4	N/A	N/A
Power8	ST, SMT2, SMT4, SMT8	SMT8	SMT8	N/A
Power9	ST, SMT2, SMT4, SMT8	SMT8	SMT8	SMT8
Power10	ST, SMT2, SMT4, SMT8	SMT8	SMT8	SMT8

**Table 1: IBM i supported and default processor threading contexts**

The default thread context is selected by the operating system, but it can be easily changed by the system administrator. For all supported releases, the default SMT level is the maximum supported by the hardware.

Compatibility mode can be changed using the Hardware Management Console (HMC) when the partition is powered down

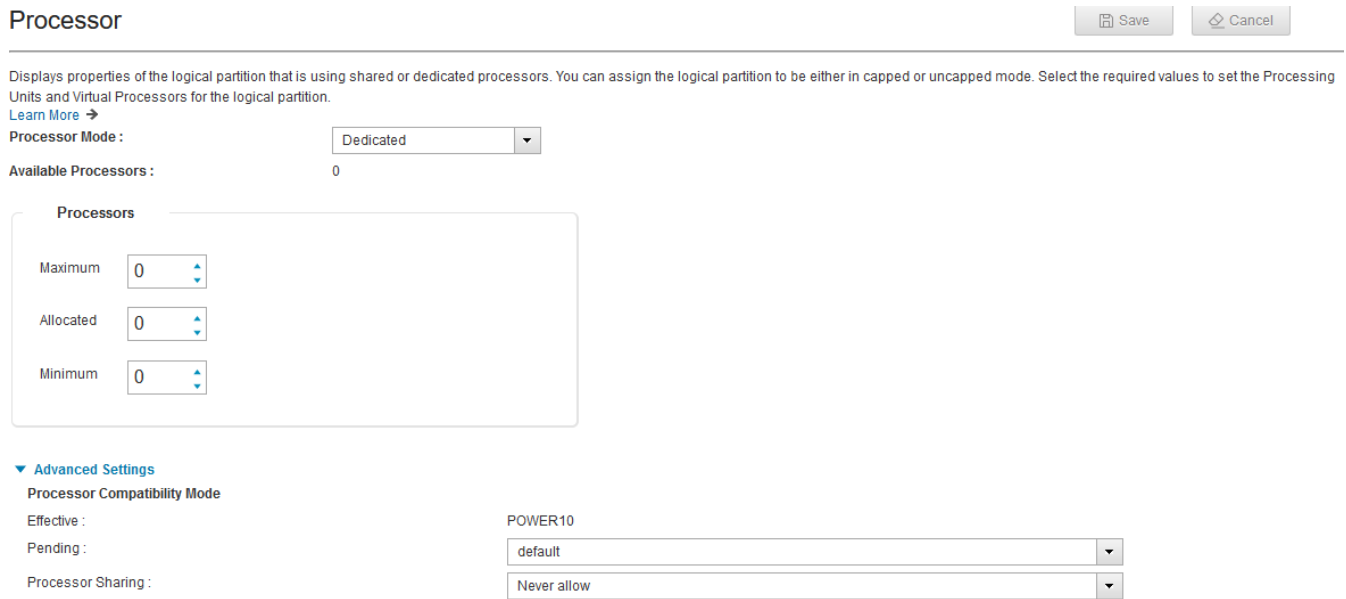


Figure 39 - Processor Compatibility Mode

On the HMC Enhanced GUI, the compatibility mode is found on the Processor tab, under Advanced Settings, after clicking on the Advanced button.

### 13.3.2 QPRCMLTTSK System Value

You can disable or enable the SMT behavior for each partition with the [Processor multitasking \(QPRCMLTTSK\) system value](#). The default for this system value is “system controlled,” which means the operating system determines the optimal setting. For Power7 systems, this is dynamic switching with four hardware threads. For Power10, Power9 and Power8 systems, this is dynamic switching with eight hardware threads.

You can explicitly turn off SMT support and always run the cores in a single-threaded (ST) mode by setting QPRCMLTTSK to \*OFF. In general, the default setting will be the best for most applications. Switching to and from single-thread (ST) context may be accomplished using the IBM i processor multitasking mode system value, QPRCMLTTSK. QPRCMLTTSK changes are effective immediately and persist across system IPL.

Supported QPRCMLTTSK values are as follows:

- 0 - Processor multitasking is disabled. This value corresponds to ST context.
- 1 - Processor multitasking is enabled. This value corresponds to SMT2 context if the partition's Processor Compatibility Mode is Power6 or Power6+. Otherwise, the thread context is determined by the maximum SMT level control.
- 2 - Processor multitasking is system controlled. This is the default value. For Power7, Power8, Power9 and Power10, the implementation is identical to '1', processor multitasking enabled.

Examples:

```
DSPSYSVAL SYSVAL(QPRCMLTTSK)
```

```
CHGSYSVAL SYSVAL(QPRCMLTTSK) VALUE('0') /* ST context */
CHGSYSVAL SYSVAL(QPRCMLTTSK) VALUE('1') /* SMTn context */
CHGSYSVAL SYSVAL(QPRCMLTTSK) VALUE('2') /* SMTn context */
```

### 13.3.3 QWCCHGPR API

The QWCCHGPR API is available to switch between the various SMT settings without rebooting the partition. When processor multitasking is enabled, switching among the thread contexts available for the partition IPL may be accomplished using the change processor multitasking information API, QWCCHGPR. QWCCHGPR API changes are effective immediately and persist across system IPL. The QWCCHGPR API takes a single parameter, the maximum number of secondary threads per processor:

- 0 – No maximum is selected. The system uses the default number of secondary threads as determined by the operating system.
- 1-255 – The system may use up to the number of secondary threads specified.

The QWCCHGPR API may be called from a command line.

Note that setting the maximum number of secondary threads does not establish the processor threading context directly. The maximum value will be accepted regardless of the processor threading contexts supported by the underlying hardware, and the OS will apply the configured maximum to the system. On a Power10, Power9 or a Power8-based system, if a maximum value is specified by the QWCCHGPR API, the OS will try to establish the maximum thread context supported as shown in Table 1, subject to the limit specified.

Examples:

```
CALL PGM(QWCCHGPR) PARM(X'00000000') /* No maximum */
CALL PGM(QWCCHGPR) PARM(X'00000001') /* SMT2 context */
CALL PGM(QWCCHGPR) PARM(X'00000003') /* SMT4 context */
CALL PGM(QWCCHGPR) PARM(X'00000007') /* SMT8 context */
```

The maximum number of secondary threads can be obtained from the retrieve processor multitasking information API (QWCRTVPR).

For more details, see the following link:

[Change Processor Multitasking Information \(QWCCHGPR\) API](#)

## 13.4 Power Savings Mode

Many customers can achieve cost savings by tuning their systems for energy management. By doing this tuning, system performance may be affected. On the positive side, tuning the energy setting allows some systems to be able to boost their processor frequency dynamically when running at lower processor utilization. On the other hand, tuning the energy setting can result in slower system performance at high utilization due to slower dynamic processor frequency changes.

Nominal energy settings provide the most consistent processor frequency behavior and allow consistent response time and throughput results. Because of this, use nominal settings if energy tuning



is not required.

Power processor-based systems include Intelligent Energy features that help to dynamically optimize energy usage and performance so that the best possible balance is maintained. Intelligent Energy features like EnergyScale(TM) dynamically optimize processor speed based on thermal conditions and system utilization. If the system's environment is favorable, the system can speed up processor cores to a frequency up to 10% higher than the normal rated speed to better handle demanding workloads. When utilization is lighter, or if environmental conditions are less favorable, the system can slow processor cores by as much as 50% to save energy. Users can select whether Intelligent Energy Optimization favors maximizing performance or energy savings based on their needs.

For more Power9 information see the following link:

[IBM EnergyScale for Power9 Processor-Based Systems](#)

For more Power10 information see the following link:

[IBM EnergyScale for Power10 Processor-Based Systems](#)

### 13.5 Adapter Placement

High-speed adapters should be placed following the recommendations of the PCI Adapter Placement recommendations for the individual Power model.

For detailed information refer to the appropriate section of the Systems Hardware information at:

[Power8 System Hardware](#)

[Power9 System Hardware](#)

[Power10 System Hardware](#)

### 13.6 Affinitized Partitions

The Power hypervisor was improved to maximize partition performance through affinitization. It optimizes the assignment of CPU and memory to partitions based on system topology. This results in a balanced configuration when running multiple partitions on a system.

When a partition is first activated, the hypervisor will allocate CPUs as close as possible to where allocated memory is located in order to reduce remote memory access. For shared partitions the hypervisor assigns a home node domain, the chip where the partition's memory is located, for each virtual processor. The hypervisor dispatches the shared partition's virtual processor(s) to run on the home node domain whenever possible. If dispatching on the home node domain is not possible due to physical processor over commitment of the system, the hypervisor will dispatch the virtual processor temporarily on another chip.

Collection Services includes metrics that can be useful for monitoring affinity activity. The primary metric to monitor is AFSCORE (affinity score) in the file QAPMSYSAFN.

iDoctor Collection Services Investigator can be used to view the placement of an IBM i partition. Right click the current Collection Services collection and select System Graphs ->Partition placement (affinity)->Partition placement (at end of collection). A report similar to the following is generated:

Interval end time	Shared LPAR	Affinity score	Book	Node	Partition processors	Partition memory (gigabytes)	Processors % of total	Memory % of total	Total processors	Total memory (gigabytes)
2015-09-16-00.0>	0	50	0000	000A	4	91.1592	12.50%	12.45%	32	732.1826
2015-09-16-00.0>	0	50	0000	000B	7	161.8076	21.87%	22.09%	32	732.1826
2015-09-16-00.0>	0	50	0000	0008	5	114.3369	15.62%	15.61%	32	732.1826
2015-09-16-00.0>	0	50	0001	000C	4	109.6865	12.50%	14.98%	32	732.1826
2015-09-16-00.0>	0	50	0001	000D	6	90.9434	18.75%	12.42%	32	732.1826
2015-09-16-00.0>	0	50	0001	000E	3	82.2822	9.37%	11.23%	32	732.1826
2015-09-16-00.0>	0	50	0001	000F	3	81.9668	9.37%	11.19%	32	732.1826

**Figure 40 - Partition Placement**

The report states whether the partition is shared or dedicated and shows which books or drawers (Book column) and chips (Node column) the partition’s processors and memory are assigned to. The report also includes an affinity score that ranges from 0 to 100, where 100 is optimal. Dynamic Logical Partitioning (DLAR) operations may cause a partition’s affinity score to decrease over time. Therefore it is a good idea to review your partition’s placement and affinity score from time to time and after DLPAR operations.

See section 14.2 for information about improving partition placement and the partition’s affinity score.

### 13.7 System Affinity Settings

IBM i provides a system value that dynamically adjusts the affinity of threads. IBM i assigns each newly created task a Home Node ID, essentially suggesting for it its preferred Power chip. Each time that a task enters a dispatchable state, the IBM i’s task dispatcher first attempts to dispatch the task on a core of this preferred node.

In the case of multithreaded jobs, it is possible to indicate either on a job basis (saying \*GROUP or \*NOGROUP on THDRSCAFN within a routing entry or prestart job entry) or system-wide (by using the QTHDRSCAFN system value) to control whether the multiple threads of a job should share the same Home Node, or whether each should be assigned an arbitrary Home Node.

- [System Affinity Setting - QTHDRSCADJ](#)
- [System Affinity Setting - Routing Entry](#)

For more details on affinity on Power hardware, see:

- [Under the Hood: NUMA on POWER7 in IBM i](#)

### 13.8 Collecting and Viewing Performance Data for Entire Physical System

IBM i customers using multiple partitions may find it very valuable to understand overall processing capability across all logical partitions on the physical system. Collection Services has the ability to collect select key performance metrics (such as CPU) for all partitions on a single server, regardless of whether the partition is running IBM i, AIX or Linux. It collects the data known to the Hypervisor and stores the data in the QAPMLPARH performance database file.

To enable the collection of this performance data, from the HMC Classic GUI Partition Properties

General tab, check the box to “Allow performance information collection”:

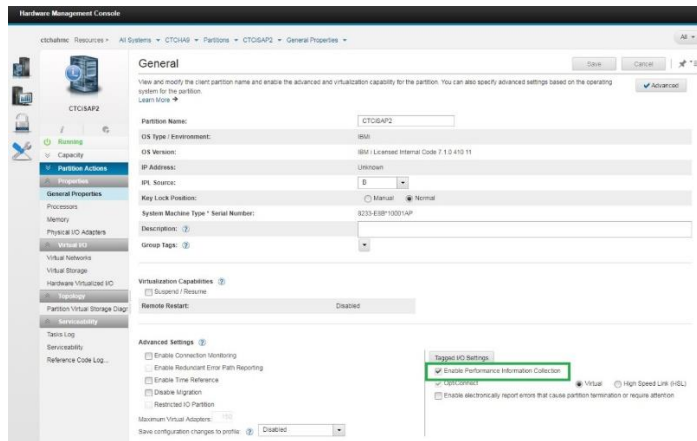


Figure 41 - Partition Properties

The setting is called “Enable Performance Information Collection” on the HMC Enhanced GUI, and is found in the Advanced Settings section of the General Properties for each partition.

You only need to enable this option on one IBM i partition. Information collected will reflect work done in partitions running AIX and Linux as well as IBM i, but only IBM i has this data collection capability.

Meaningful charts are available in both PDI and iDoctor to visualize the data.

In PDI, navigate to Investigate Data and the Collection Services package. The following charts are available, as well as more not shown:

Package Name ↑↓	Perspective ↑↓	Description ↑↓
Collection Services	Filter	processor
Collection Services	<a href="#">Logical Partitions Overview</a>	including operating system, number of virtual processors, partition memory, donated CPU time, uncapped CPU time used and others.
Collection Services	<a href="#">Donated Processor Time by Logical Partition</a>	This chart shows the processor time that has been donated by dedicated processor logical partitions that are configured to donate unused processor cycles.
Collection Services	<a href="#">Uncapped Processor Time Used by Logical Partition</a>	This chart shows the uncapped processor time that has been used by logical partitions in excess of their entitled processing capacity configured.
Collection Services	<a href="#">Virtual Shared Processor Pool Utilization</a>	This chart shows processing capacity available and processing capacity utilization for virtual shared processor pools.
Collection Services	<a href="#">Dedicated Processors Utilization by Logical Partition</a>	This chart shows the dedicated processor utilization by logical partitions.
Collection Services	<a href="#">Physical Processors Utilization by Processor Status Overview</a>	This chart shows a summary of processors utilization for the entire collection identifying utilization for dedicated processors and for shared processors.
Collection Services	<a href="#">Physical Processors Utilization by Processor Status Detail</a>	This chart shows processors utilization over time identifying utilization for dedicated processors and for shared processors.

Figure 42 - PDI Physical System Perspectives

# IBM Power Performance

Below is an example of the Logical Partitions Overview perspective showing various CPU related metrics across several partitions:

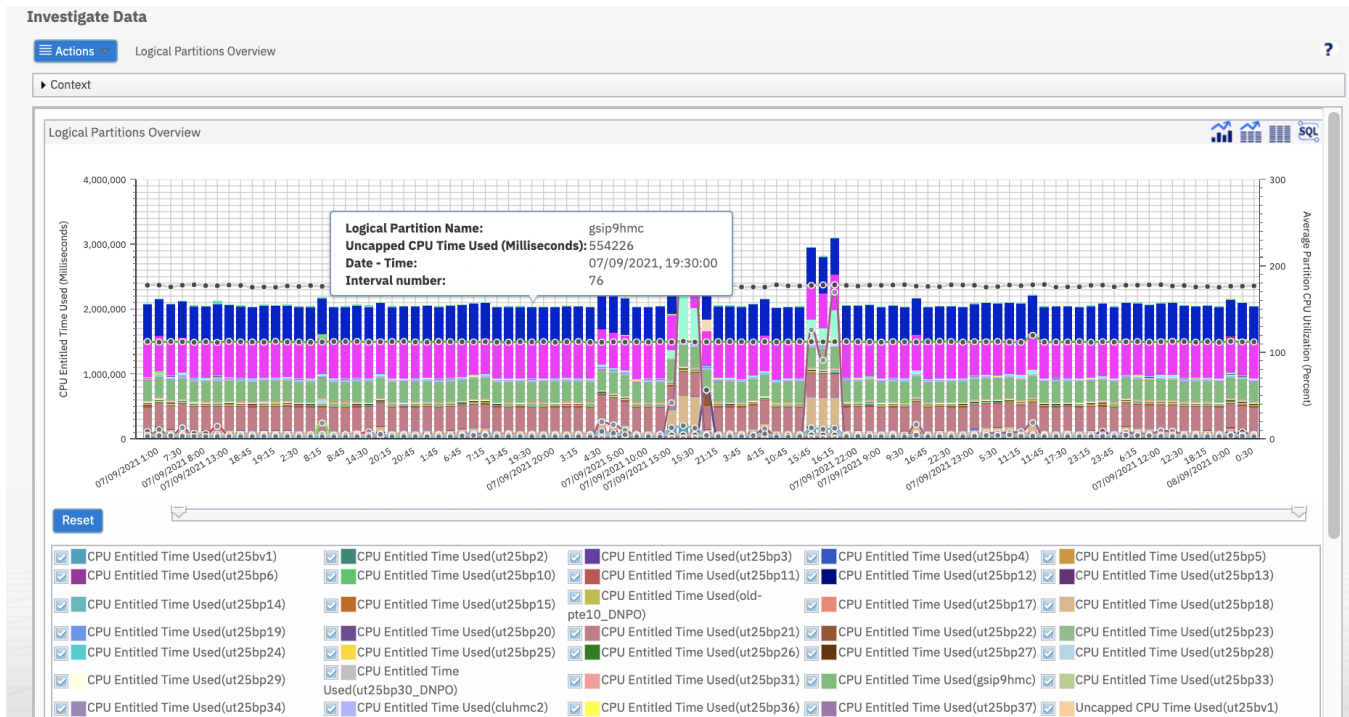


Figure 43 - Logical Partitions Overview

In iDoctor, many charts are available in Collections Services Investigator under the System graphs folder:

- System graphs
  - LPAR graphs by OS type
  - LPAR graphs by shared processor status
  - LPAR graphs by virtual shared pool ID
  - TLBIEs
  - Rankings
  - Partition placement (affinity)

Figure 44 - iDoctor System Graphs

For additional information:

[IBM Docs: Collecting and displaying CPU for all partitions](#)

### 13.8.1 Viewing Performance Data for Multiple Systems

#### 13.8.1.1 Prometheus and Grafana

[Prometheus](#) and [Grafana](#) are leading open source monitoring solutions. Prometheus delivers a time-series database to keep track of historical data and is often paired with Grafana for dashboards and visualization.

A Prometheus exporter is available for IBM i and other databases. It provides an interface for passive metrics collection. That is, Prometheus can scrape this exporter for metrics. It exports over 400 metrics (including Apache HTTP server metrics), is built around SQL and designed to be user-extensible, and lets you connect from several other monitoring stacks.

Additional information on the exporter, including installation and metrics, can be found here: <https://github.com/ThePrez/prometheus-exporter-jdbc>



Figure 45 - Prometheus visualization with Grafana

#### 13.8.1.2 Instana

Instana provides an industry-leading enterprise observability and monitoring solution, including capability for IBM i metrics.

Additional information regarding Instana is available at:

[https://www.instana.com/docs/using\\_instana/](https://www.instana.com/docs/using_instana/)

Links to IBM i OS and Db2 on i metrics available:

<https://www.instana.com/docs/ecosystem/ibmios/>

<https://www.instana.com/docs/ecosystem/ibmidb2/>

In addition, since the Prometheus protocol is so prevalent, it is understood by many enterprise monitoring tools, including Instana.

Additional information regarding this integration can be found here:

<https://techchannel.com/open-source-on-ibm-i/monitoring-ibm-i-with-prometheus/>

Link to additional information on the Prometheus Exporter for IBM i can be found here:

<https://github.com/ThePrez/prometheus-exporter-jdbc>

### 13.8.1.3 *Nagios*

Nagios provides enterprise-class Open Source IT monitoring, network monitoring, server and applications monitoring. Multiple systems can be monitored in a central view. An IBM i plug-in is available, which provides monitoring and alerting capability for several key i metrics.

Additional information regarding Nagios is available at:

<https://www.nagios.org/>

In addition, since the Prometheus protocol is so prevalent, it is understood by many enterprise monitoring tools, including Nagios.

Additional information regarding this integration can be found here:

<https://techchannel.com/open-source-on-ibm-i/monitoring-ibm-i-with-prometheus/>

Link to additional information on the Prometheus Exporter for IBM i can be found here:

<https://github.com/ThePrez/prometheus-exporter-jdbc>

## 13.9 How does IBM i report CPU utilization for SMT processors?

For historical reasons, processor utilization is usually calculated in the time domain. The processors used in early time-sharing computer systems were fixed-frequency and single-threaded; processing capacity per unit of time was constant. Because processor activity alternated between idle and run states, it was convenient to express processor utilization as the percentage of time that the processor was utilized (i.e. run-state) over an interval of interest.

$$\text{PROCESSOR UTILIZATION (\%)} = 100 * \text{RUN-STATE TIME} / \text{AVAILABLE TIME}$$

### Equation 1

When Power processors employed SMT, it was no longer possible to accurately report processor utilization based on the percentage of time that the processor was in run state. Consider the case of Power8 in the SMT8 context. When all processor threads are idle, utilization should be 0%. When no

processor threads are idle, utilization should be 100%. What should processor utilization be for all other combinations of idle and run state? Let's consider some options:

1. Treat the processor as idle (0% utilization). This is obviously wrong.
2. Continue to treat the processor as non-idle (100% utilization). While this option may be appropriate from the system context, because a processor dispatched to one partition cannot be simultaneously dispatched to another, it is not ideal within the partition context, because it causes utilization to be grossly over-reported; it does not recognize the capacity available in the idle threads that is available to the partition to which the processor is dispatched.
3. Treat processor threads as individual processors and report processor-thread based utilization in the time domain. This causes utilization to be under-reported because SMT efficiencies are much less than ideal. For example, when half of the threads are idle, there is much less than 50% additional capacity available in the processor.

To address this issue, the Power processor created the **Processor Utilization Resource Register (PURR)** with following properties:

- The PURR is defined such that  $\sum (\Delta \text{PURR}) = \Delta \text{TIME}$  over any interval. That is, the PURR apportions processor time among the processor's threads.
- The PURR assigned to the set of idle threads can be calibrated for combinations of idle and run state threads for each SMT context.

With these properties and with PURR virtualization by the PowerVM hypervisor, processor utilization can continue to be reported in the time domain for shared and dedicated processors alike. The PURR is what allows IBM i to report processor utilization that accurately reflects the additional compute capacity remaining in the SMT core. In terms of PURR and TIME, **Equation 1** becomes:

$$\text{PROCESSOR UTILIZATION (\%)} = 100 * \text{NON-IDLE PURR} / \text{AVAILABLE TIME}$$

**Equation 2**

**Equation 2** is the standard partition processor utilization metric reported by WRKSYSACT and other IBM i performance tools.

### 13.10 How is IBM i CPU utilization calibrated for SMT processors?

The standard IBM i processor utilization metric is based on the Processor Utilization Resource Register (PURR) of the Power processor. The PURR is defined in [section 13.9](#).

PURR calibration is the key feature that enables IBM i processor utilization to accurately reflect the SMT scaling characteristics of a specific processor implementation. For example, the following graph illustrates the difference in PURR calibration derived from Power8, Power9, and Power10 systems in SMT8 context:



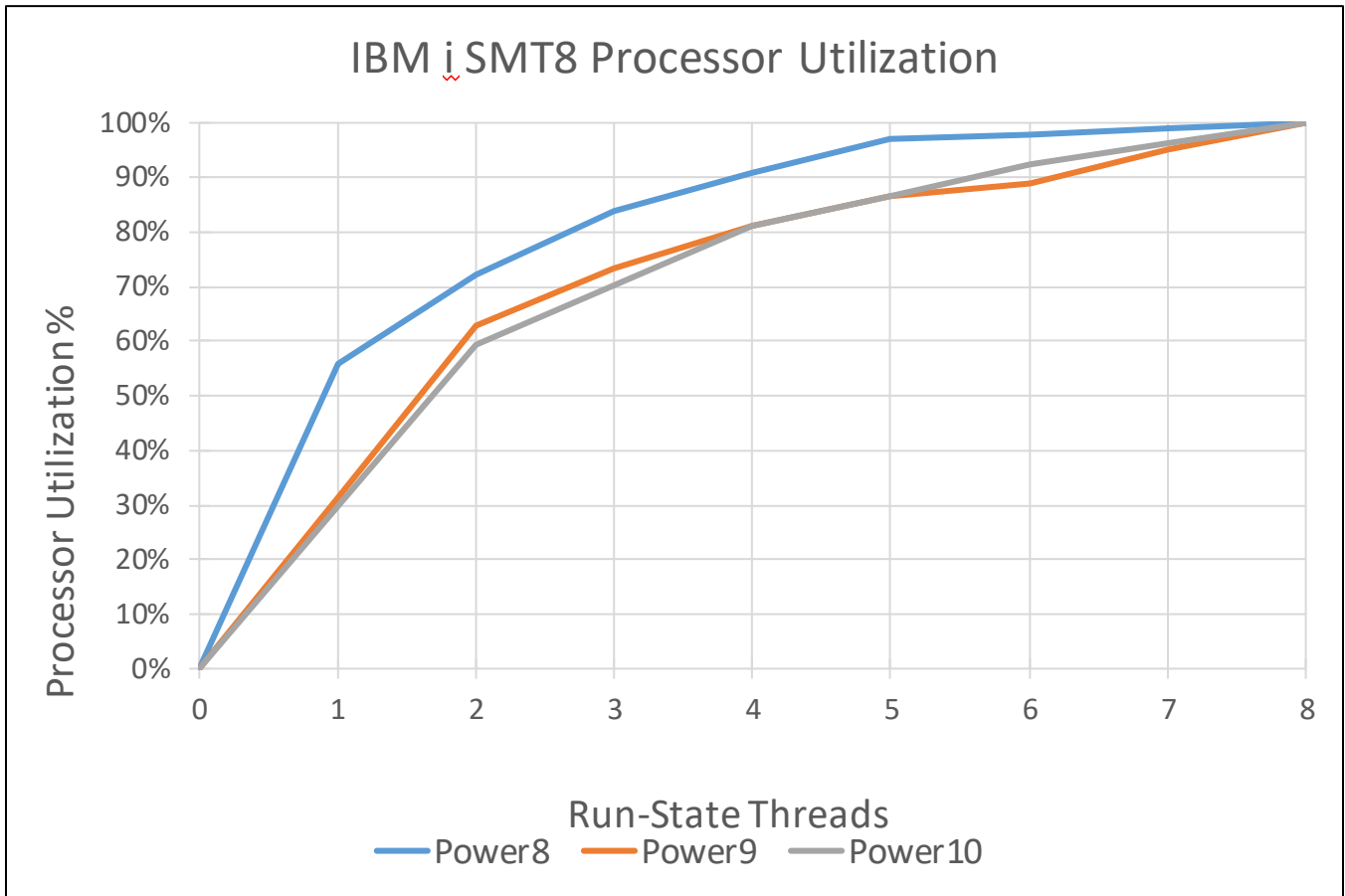


Figure 46 - IBM i SMT8 CPU Utilization Reporting

IBM uses same workloads for PURR calibration as those used to establish rPERF and CPW ratings, the objective being that the IBM i processor utilization metric accurately represents the percent of rated capacity utilized throughout its entire range. Ultimately, the IBM i processor utilization metric is an approximation, and actual computational utilization may be influenced by numerous factors such as workload, partition configuration, and energy management.

### 13.11 Power8 Redbook

The Power8 Performance Redbook is available at:

[Performance Optimization and Tuning Techniques for IBM Processors, including IBM Power8](#)

### 13.12 Power9 and Power10 Redbooks

The Power10 Performance Redbooks are available at:

[IBM Power Systems Redbooks](#)

## 14 Virtualization

### 14.1 Virtualization Best Practices

This section summarizes the Power Virtualization Best Practices. For detailed description and latest updates, refer to the latest version of the IBM Power Virtualization Best Practices Guide at the following link:

[IBM Power Virtualization Best Practice Guide](#)

Information on how shared processor performance compares to dedicated processors is available in this blog entry in the PowerVM Community:

[How does Shared Processor Performance Compare to Dedicated Processors](#)

Detailed information about Power7 partitions and virtualization is available in the following white paper:

[Under the Hood: Power7 Logical Partitions](#)

Other excellent resources for details on PowerVM are available in the following links:

[PowerVM Processor Virtualization 101](#)

[PowerVM Processor Virtualization 101 Part 2](#)

### 14.2 Dynamic Platform Optimizer (DPO)

The Dynamic Platform Optimizer (DPO) can be used to report affinity scores for partitions and to potentially improve partition placement for partitions with poor placement. On recent firmware DPO can be used to report the affinity score of individual partitions:

```
➤ lsmemopt -m <managed_system> -o currscore -r lpar
lpar_name=mylpar1,lpar_id=1,curr_lpar_score=100
lpar_name=mylpar2,lpar_id=2,curr_lpar_score=80
```

DPO can also be used to predict potential placement improvements:

```
➤ lsmemopt -m <managed_system> -o calcscore -r lpar
lpar_name=mylpar1,lpar_id=1,curr_lpar_score=100,predicted_lpar_score=100
lpar_name=mylpar2,lpar_id=2,curr_lpar_score=80,predicted_lpar_score=100
```

DPO can be used to re-assign partition resources, potentially improving affinity and partition performance. However, DPO may reduce partition performance while it is actively moving resources. Refer to the following resource before using DPO:

[Dynamic Platform Optimizer](#)

### 14.3 VIOS Advisor

The VIOS advisor is an application that runs within the customer's VIOS for a user specified amount of time (hours). It polls and collects key performance metrics before analyzing results and providing a health check report. Additionally, it proposes changes to the environment or areas to investigate further.

## IBM Power Performance

The goal of the VIOS advisor is to be an expert system that analyzes performance metrics and makes assessments and recommendations based on the expertise and experience available within the IBM systems performance group.

Additional VIOS Advisor information can be found at the following website:

[VIOS Performance Advisor](#)

## 15 Reporting a Performance Problem

### 15.1 Introduction

The focus of this chapter is on what information and data should be provided when reporting a performance problem to post-sales support. However, the information provided in this chapter also applies when reporting performance problems to pre-sales and service organizations.

### 15.2 Define the Performance Problem

The first question support will ask when reporting a performance problem is, “What is slow?” A good understanding about what exactly is slow helps the performance analyst concentrate the analysis on the component(s) that is most likely causing the performance problem.

As a rule, the better the description of the performance problem the more likely there will be a quick identification of the root cause and potential resolution.

The IBM i Global Support Center (iGSC) has provided several Must Gather Documents with detailed instructions for gathering the data required to analyze performance problems.

[Master Document List for Performance](#)

Over the last several years the iGSC has worked on automating the majority of the Must Gather Documents into a suite of tools known as QMGTOOLS. Many of the Must Gather Documents have been updated to point at QMGTOOLS functions. When working with support, you will likely be asked to use QMGTOOLS in order to automate the collection of data needed to assist in debug. As a proactive measure, it is recommended to ensure the QMGTOOLS library is installed on your system and is periodically updated. Instructions are available here: [How To Obtain and Install QMGTOOLS](#)

You may also be referred to the iDoctor website in order to utilize iDoctor. If iDoctor is needed by IBM Support to assist with aCase, then a temporary license key will be provided.

<http://public.dhe.ibm.com/services/us/igsc/idoctor/html/downloadOptions.html>

### 15.3 Questions that help IBM diagnose the problem

The following is a list of questions that help to define a performance problem:

- Can you append more detail on the simplest, repeatable example of the problem?
- Can the problem be demonstrated with the execution of a specific command or sequence of events?
- Were any new IBM PTFs or groups applied just before the problem starting?
- Does the problem impact interactive users, batch jobs, or both?
- What cumulative PTF package is installed on the system?
- What is the level of the hiper, database, Java, tech refresh, and other applicable group PTFs?
- Were any application changes or vendor fixes applied recently?
- Has the CPU utilization changed since the problem started? Is it higher or lower than it was before?
- How has the disk utilization changed since the problem started?
- If the problem impacts a specific job, user, or subset of jobs, describe these in the greatest

detail possible.

- When do any problem jobs or problem workloads run?
- Does the performance problem involve SQL access to the database? If so, has a specific statement or type of SQL statement been identified? Has the SQL been analyzed with Visual Explain and indexed properly?
- Were there any hardware upgrades or changes since the problem happened?
- Is there any performance data available from before the problem started? If not, is there any previous data that can be retrieved from tape?

## 16 References

The following sections provide links to various types of online resource information and documentation for IBM i:

### 16.1 Blogs on Performance Topics

Useful links that contain many performance-related posts:

- PowerVM (IBM Power Community): [PowerVM](#)
- Db2 for IBM i <https://Db2IBMi.blogspot.com/>
- IBM iSee Video Blog (useful Systems Management and Performance-related demonstrations) : [iSee Video Blog](#)
- i Can blog by Dawn May (Technical Tips for i): <https://dawnmayi.com/i-can-blog-of-blogs/>

### 16.2 Education and Training

The IBM Technology Expert Labs performance team can do customized formal or informal education or skills transfer on any IBM i performance-related topic. Contact the IBM i performance team leader, Eric Barsness, at [ericbar@us.ibm.com](mailto:ericbar@us.ibm.com) for more information. For knowledge and skills transfer sessions that cover Db2 for i, SQL, and data-centric application performance / scalability monitoring, analysis and tuning contact Kent Milligan at [kmill@us.ibm.com](mailto:kmill@us.ibm.com).

### 16.3 Redbooks and Redpapers

The following are selected Redbooks related to IBM i Performance Tools:

- End to End Performance Management on IBM i  
<http://www.redbooks.ibm.com/abstracts/sg247808.html>
- IBM eServer iSeries Performance Management Tools  
<http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/redp4026.html?Open>

While the following Redbooks were written in the V5R4 timeframe, the majority of their technical content is still applicable:

- [Application and Program Performance Analysis Using PEX Statistics on IBM i5/OS](#)
- [IBM iDoctor iSeries Job Watcher: Advanced Performance Tool](#)
- [Best Practices for Managing IBM i Jobs](#)
- [i5/OS Diagnostic Tools for System Administrators: An A to Z Reference for Problem Determination](#)

The following are selected Redbooks related to IBM Power Performance and Virtualization:

- IBM PowerVM Virtualization Introduction and Configuration  
<http://www.redbooks.ibm.com/abstracts/sg247940.html>
- IBM PowerVM Best Practices

<http://www.redbooks.ibm.com/abstracts/sg248062.html>

- IBM PowerVM Virtualization Managing and Monitoring  
<http://www.redbooks.ibm.com/abstracts/sg247590.html>

## 16.4 Additional Performance-related information

An internet search is a common and valid way of searching for IBM i information, and there is plenty of it out there. The following are selected performance-related links to key topics:

- IBM Workload Estimator: [WLE](#)
- IBM i Performance Tools: [IBM Support: Performance Tools](#)
- IBM i Performance Data Investigator: [IBM Support: PDI](#)
- IBM iDoctor for IBM i: [iDoctor](#)
- IBM i Wait Accounting information:
  - [Job Waits Whitepaper](#)
  - [The basics of Wait Accounting](#)
  - [IBM i Wait Accounting](#)
- Best Practices for Managing IBM i Jobs and Output: [Jobs Redpaper](#)
- IBM Tivoli Monitoring: [Tivoli Monitoring](#)
- SSD references:
  - [Concurrently Move Db2 for i Tables and Indexes to Solid State Disks](#)
  - [Customer Use of SSD \(Solid State Drives\)](#)

## 16.5 Database Performance Articles

- Db2 for i– “Database performance and query optimization”  
[Database Performance and Query Optimization](#)
- DDS and SQL - The Winning Combination for Db2 for i: Business and technical benefits of a modern Db2 for i database  
[DDS and SQL - The Winning Combination for Db2 for i](#)
- Get Your System Humming: 7 Great Tips for Tuning Db2 for i  
[Get Your System Humming: 7 Great Tips for Tuning Db2 for i](#)
- Process your Db2 for i indexes in parallel  
[Process your Db2 for i indexes in parallel](#)

## 16.6 Key Performance Resources

The following are valuable resources to use as starting points when researching performance-related topics:

- IBM i Performance and Capacity  
[Performance and Capacity](#)
- IBM 7.5 Docs:  
[7.5 Docs: Performance Topic](#)
- IBM 7.4 Docs:  
[7.4 Docs: Performance Topic](#)
- IBM 7.3 Docs:  
[7.3 Docs: Performance Topic](#)
- IBM Systems Hardware (IBM Docs):

[Power8](#)

[Power9](#)

[Power10](#)

- IBM Power: Power10 Performance Quick Start Guides  
[Power10 Performance Quick Start Guides](#)
- IBM Developer – IBM i:  
<https://developer.ibm.com/components/ibm-i/>
- IBM i Support Software Knowledge Base:  
[Software Knowledge Base](#)

## **16.7 Performance Capabilities Reference Note**

The Performance Capabilities Reference Manual (PCRM) was formerly the reference manual for all things related to IBM i performance considerations. Beginning in 2014, the PCRM only covers CPW rating information. This includes updates for new hardware models and CPW ratings. Older versions are still available for download. For detailed IBM i performance information, use the links referenced above.

[IBM i Performance Capabilities Reference](#)





The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

UNIX is a registered trademark of The Open Group in the United States, other countries or both.

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

TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).

SPECint, SPECfp, SPECintb, SPECweb, SPECAppServer, SPEC OMP, SPECviewperf, SPECcapc, SPECchpc, SPECjvm, SPECmail, SPECimaps and SPECsfs are trademarks of the Standard Performance Evaluation Corporation (SPEC).

InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

© IBM Corporation 2024

IBM Corporation  
New Orchard Road  
Armonk, NY 10504  
U.S.A.

Produced in the United States of America May 2024

All Rights Reserved

This document was developed for products and/or services offered in the United States. IBM might not offer the products, features, or services discussed in this document in other countries.

The information is subject to change without notice. Consult your local IBM business contact for information on the products, features, and services available in your area.

All statements regarding IBM future directions and intent are subject to change or withdrawal without notice and represent goals and objectives only.

IBM, the IBM logo, ibm.com, AIX, Power Systems, Power7, Power7+, Power8, Power9, Power10, TurboCore and Active Memory are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. 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 "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml)

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

IBM hardware products are manufactured from new parts, or new and used parts. In some cases, the hardware product may not be new and may have been previously installed. Regardless, our warranty terms apply.

Photographs show engineering and design models. Changes may be incorporated in production models.

Copying or downloading the images contained in this document is expressly prohibited without the written consent of IBM.

This equipment is subject to FCC rules. It will comply with the appropriate FCC rules before final delivery to the buyer.

Information concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of the non-IBM products should be addressed with those suppliers.

All performance information was determined in a controlled environment. Actual results may vary. Performance information is provided "AS IS" and no warranties or guarantees are expressed or implied by IBM. Buyers should consult other sources of information, including system benchmarks, to evaluate the performance of a system they are considering buying.

When referring to storage capacity, 1 TB equals total GB divided by 1000; accessible capacity may be less.

The IBM home page on the Internet can be found at:

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

A full list of U.S. trademarks owned by IBM may be found at:

<https://www.ibm.com/legal/copytrade>.

The IBM Power home page on the Internet can be found at: [IBM Power Systems Servers](#)

POW03102-USEN-23