IBM Power Systems
Performance Capabilities Reference
IBM i operating system 7.1

February 2013

This document is intended for use by qualified performance related programmers or analysts from IBM, IBM Business Partners and IBM customers using the IBM Power™ Systems platform running IBM i operating system. Information in this document may be readily shared with IBM i customers to understand the performance and tuning factors in IBM i operating system 7.1 and earlier where applicable. **For the latest updates and for the latest on IBM i performance information, please refer to the Performance Management Website: http://www.ibm.com/systems/power/software/i/management/performance/index.html**

Requests for use of performance information by the technical trade press or consultants should be directed to STG Cross Platform Systems Performance Department.

Note!

Before using this information, be sure to read the general information under “Special Notices.”

Thirty-sixth Edition (February 2013) SC41-0607-15

This edition applies to IBM i operating system 7.1 running on IBM Power Systems.

You can access this document for download from the IBM Performance Management on IBM i web site at: http://www-03.ibm.com/systems/power/software/i/management/performance/resources.html .

The document is viewable/downloadable in Adobe Acrobat (.pdf) format and is approximately 1.5 MB in size. Adobe Acrobat reader plug-in is available at: http://www.adobe.com .

Table of Contents

Special Notices

DISCLAIMER NOTICE

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. This information is presented along with general recommendations to assist the reader to have a better understanding of IBM(*) products. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described within the document and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

Information is provided "AS IS" without warranty of any kind.

The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide home pages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

The following terms, which may or may not be denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation.

iSeries or AS/400	System/370	Operating System/400
C/400	IPDS	i5/OS
OS/400	COBOL/400	Application System/400
System i5	RPG/400	OfficeVision
System i	IBM i operating system	Facsimile Support/400
PS/2	DRDA	Distributed Relational Database Architecture
OS/2	SQL/400	Advanced Function Printing
DB2	ImagePlus	Operational Assistant
AFP	VTAM	Client Series
IBM	APPN	Workstation Remote IPL/400
SQL/DS	SystemView	Advanced Peer-to-Peer Networking
400	ValuePoint	OfficeVision/400
CICS	DB2/400	iSeries Advanced Application Architecture
S/370	ADSM/400	ADSTAR Distributed Storage Manager/400
RPG IV	AnyNet/400	IBM Network Station
AIX	IBM XIV Storage systems	Lotus, Lotus Notes, Lotus Word Pro, Lotus 1-2-3
Micro-partitioning	POWER4	POWER4+
POWER	POWER5	POWER5+
Power™ Systems	POWER6	POWER6+
PowerPC	POWER7	Power™ Systems Software
IBM PureSystems	IBM PureFlex System	IBM PureApplication System

The following terms, which may or may not be denoted by a double asterisk (**) in this publication, are trademarks or registered trademarks of other companies as follows:

TPC Benchmark	Transaction Processing Performance Council
TPC-A, TPC-B	Transaction Processing Performance Council
TPC-C, TPC-D	Transaction Processing Performance Council
ODBC, Windows NT Server, Access	Microsoft Corporation
Visual Basic, Visual C++	Microsoft Corporation
Adobe PageMaker	Adobe Systems Incorporated
Borland Paradox	Borland International Incorporated
CorelDRAW!	Corel Corporation
Paradox	Borland International
WordPerfect	Satelite Software International
BEST/1	BGS Systems, Inc.
NetWare	Novell
Compaq	Compaq Computer Corporation
Proliant	Compaq Computer Corporation
BAPCo	Business Application Performance Corporation
Harvard	Gaphics Software Publishing Corporation
HP-UX	Hewlett Packard Corporation
HP 9000	Hewlett Packard Corporation
INTERSOLV	Intersolve, Inc.
Q+E	Intersolve, Inc.
Netware	Novell, Inc.
SPEC	Systems Performance Evaluation Cooperative
UNIX	UNIX Systems Laboratories
WordPerfect	WordPerfect Corporation
Powerbuilder	Powersoft Corporation
SQLWindows	Gupta Corporation
NetBench	Ziff-Davis Publishing Company
DEC Alpha	Digital Equipment Corporation

Microsoft, Windows, Windows 95, Windows NT, Internet Explorer, Word, Excel, and Powerpoint, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Other company, product or service names may be trademarks or service marks of others.

Purpose of this Document

The purpose of this document is to help provide guidance in terms of IBM i operating system performance, capacity planning information, and tips to obtain optimal performance on IBM i operating system.

This document is typically updated with each new release or more often if needed. This February 2013 edition of the IBM i 7.1 Performance Capabilities Reference Guide is an update to the November 2012 edition to reflect new product functions announced on February 5, 2013.

This edition includes performance information on newly announced POWER7+ models including the Power 710, 720, 730, 740, 750, and 760. CPW values are included for the POWER7+ based Compute Nodes for the IBM Pure Flex System p260, and CPW values are also provided for the IBM POWER 770 and 780 models announced in October 2012 which include the 9117-MMD and 9179-MHD, both using POWER7+ technology.

This document also includes performance information on IBM Power Systems featuring POWER7 processor technology. The Power 710, 720, 730, 740, 750 Express, Power 770, Power 780 and Power 795 offer a broad range of capacity and performance. This document further includes information on DB2 UDB for iSeries SQL Query Engine Support, Solid State Drives (SSDs), Websphere Application Server including WAS V6.1 both with the Classic VM and the IBM Technology for Java (32-bit) VM, WebSphere Host Access Transformation Services (HATS) including the IBM WebFacing Deployment Tool with HATS Technology (WDHT), PHP - Zend Core for i, Java including IBM Technology for Java 32-bit and IBM Technology for Java 64-bit, Domino 8.5, new internal storage adapters, DASD IO performance for the Power 750 and Power 770 models, Virtual Tape, IPL performance, Energy Management (including discussions of Dynamic Power Save mode) and Simultaneous Multi-Threading (SMT).

The wide variety of applications available makes it extremely difficult to describe a "typical" workload. The data in this document is the result of measuring or modeling certain application programs in very specific and unique configurations, and should not be used to predict specific performance for other applications. The performance of other applications can be predicted using a system sizing tool such as IBM Systems Workload Estimator (refer to Chapter 20 for more details on the Workload Estimator).

Chapter 1. Introduction

IBM System i and IBM System p platforms unified the value of their servers into a single, powerful lineup of servers based on industry leading Power Systems processor technology with support for IBM i operating system (formerly known as i5/OS), IBM AIX and Linux for Power.

Following along with this exciting unification are a number of naming changes to the formerly named i5/OS, now officially called IBM i operating system. Specifically, recent versions of the operating system are referred to by IBM i operating system 7.1, IBM i operating system 6.1 (previously i5/OS V6R1), and IBM i operating system 5.4 (previously i5/OS V5R4). Shortened forms of the latest operating system name are IBM i 7.1, i 7.1, i V7.1 iV7R1, and sometimes simply 'i'. As always, references to legacy hardware and software will commonly use the naming conventions of the time.

IBM PureSystems and IBM Power Systems running POWER7 and POWER7+ technology are designed to deliver unprecedented performance, scalability, reliability and manageability for demanding commercial workloads. With offerings starting at 4 cores, the Power 710, 720, 730, 740, 750 express, Power 760, Power 770, Power 780, Power 795 each offer 64-bit architecture and include up to eight cores on a single-chip module (SCM), and contain 2 MB of L2 cache (256 KB per core) and 32 MB of L3 cache (4 MB per core).

The Power 780 and Power 795 possess the ability to switch between its standard throughput optimized mode and its unique TurboCore mode, where performance per core is boosted with access to both additional cache and additional clock speed. Based on the user's configuration option, any Power 780 system can be booted in standard mode, enabling up to a maximum of 64 processor cores running at 3.86 GHz, or in TurboCore mode, enabling up to 32 processor cores running at 4.14 GHz and twice the cache per core. The Power 795 runs with frequency of 4.0 GHz in normal boot mode, and 4.25 GHz in TurboCore mode. Please reference Appendix C for more details.

Customers who wish to remain with their existing hardware but want to move to IBM i 7.1 may find functional and performance improvements. IBM i 7.1 continues to help protect the customer's investment while providing more function and better price/performance over previous versions. The primary public performance information web site is found at:
http://www.ibm.com/systems/power/software/i/management/performance/index.html

Chapter 2. Communications Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

There are many factors that affect System i performance in a communications environment. This chapter discusses some of the common factors and offers guidance on how to help achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and analysis with the NetPerf and Netop workloads. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics. The NetPerf and Netop workloads are described in section 2.2.

This chapter focuses on communication in non-secure and secure environments on Ethernet solutions using TCP/IP. Many applications require network communications to be secure. Communications and cryptography, in these cases, must be considered together. Secure Socket Layer (SSL), Transport Layer Security (TLS) and Virtual Private Networking (VPN) capacity characteristics will be discussed in section 2.5 of this chapter. For information about how the Cryptographic Coprocessor improves performance on SSL/TLS connections, see section 3.4 of Chapter 3, “Cryptography Performance.”

Communications Performance Highlights for IBM i Operation System 5.4:

- The support for the new Internet Protocol version 6 (IPv6) has been enhanced. The new IPv6 functions are consistent at the product level with their respective IPv4 counterparts.

- Support is added for the 10 Gigabit Ethernet optical fiber input/output adapters (IOAs) 573A and 576A. These IOAs do not require an input/output processor (IOP) to be installed in conjunction with the IOA. Instead the IOA can be plugged into a PCI bus slot and the IOA is controlled by the main processor. The 573A is a 10 Gigabit SR (short reach) adapter, which uses multimode fiber (MMF) and has a duplex LC connector. The 573A can transmit to lengths of 300 meters. The 576A is a 10 Gigabit LR (long reach) adapter, which uses single mode fiber (SMF) and has a duplex SC connector. The 576A can transmit to lengths of 10 kilometers. Both of these adapters support TCP/IP, 9000-byte jumbo frames, checksum offloading and the IEEE 802.3ae standard.

- The IBM 5706 2-Port 10/100/1000 Base-TX PCI-X IOA and IBM 5707 2-Port Gigabit Ethernet-SX PCI-X IOA supports checksum offloading and 9000-byte jumbo frames (1 Gigabit only). These adapters do not require an IOP to be installed in conjunction with the IOA.

- The IBM 5701 10/100/1000 Base-TX PCI-X IOA does not require an IOP to be installed in conjunction with the IOA.

- The IBM Cryptographic Access Provider product, 5722-AC3 (128-bit) is no longer required. This is a new development for the 5.4 release of IBM i Operation System. All 5.4 systems are capable of the function that was previously provided in the 5722-AC3 product. This is relevant for SSL communications.

Communications Performance Highlights for IBM i Operation System 5.4.5:

- The IBM 5767 2-Port 10/100/1000 Based-TX PCI-E IOA and IBM 5768 2-Port Gigabit Ethernet-SX PCI-E IOA supports checksum offloading and 9000-byte jumbo frames (1 Gigabit only). These adapters do not require an IOP to be installed in conjunction with the IOA.

- IBM's Host Ethernet Adapter (HEA) integrated 2-Port 10/100/1000 Based-TX PCI-E IOA supports checksum offloading, 9000-byte jumbo frames (1 Gigabit only) and LSO - Large Send Offload (IPv4 only). These adapters do not require an IOP to be installed in conjunction with the IOA. Additionally, each physical port has 16 logical ports that may be assigned to other partitions and allows each partition to utilize the same physical port simultaneously with the following limitation: one logical port, per physical port, per partition.

Communications Performance Highlights for IBM i Operation System 6.1:

- Additional enhancement in Internet Protocol version 6 (IPv6) in the following areas:
  1. Advanced Sockets APIs
  2. Path MTU Discovery
  3. Correspondent Node Mobility Support
  4. Support of Privacy extensions to stateless address auto-configuration
  5. Virtual IP address,
  6. Multicast Listener Discovery v2 support
  7. Router preferences and more specific route advertisement support
  8. Router load sharing.

- Additional enhancement in Internet Protocol version 4 (IPv4) in the following areas:
  1. Remote access proxy fault tolerance
  2. IGMP v3 support for IPv4 multicast.

- Large Send Offload support was implemented for Host Ethernet Adapter ports on Internet Protocol version 4 (IPv4).

2.1 System i Ethernet Solutions

The need for communication between computer systems has grown over the last decades, and TCP/IP over Ethernet has grown with it. We currently have arrived where different factors influence the capabilities of the Ethernet. Some of these influences can come from the cabling and adapter type chosen. Limiting factors can be the capabilities of the hub or switch used, the frame size you are able to transmit and receive, and the type of connection used. The System i server is capable of transmitting and receiving data at speeds of 10 megabits per second (10 Mbps) to 10 gigabits per second (10 Gbps or 10 000 Mbps) using an Ethernet IOA. Functions such as full duplex also enhance the communication speeds and the overall performance of Ethernet.

Table 2.1 contains a list of Ethernet input/output adapters that are used to create the results in this chapter.

Table 2.1

Ethernet input/output adapters

CCIN[3]	Description	Speed[6] (Mbps)	Jumbo frames supported	Operations Console supported	Duplex mode capability	
					Full	Half
2849[1]	10/100 Mbps Ethernet	10 / 100	No	Yes	Yes	Yes
5700[2]	IBM Gigabit Ethernet-SX PCI-X	1000	Yes	No	Yes	No

5701[1]	IBM 10/100/1000 Base-TX PCI-X	10 / 100 / 1000	Yes	No	Yes	Yes
5706[1]	IBM 2-Port 10/100/1000 Base-TX PCI-X[7]	10 / 100 / 1000	Yes	Yes	Yes	Yes
5707[2]	IBM 2-Port Gigabit Ethernet-SX PCI-X[7]	1000	Yes	Yes	Yes	No
5767[1]	IBM 2-Port 10/100/1000 Base-TX PCI-e[7]	10 / 100 / 1000	Yes	Yes	Yes	Yes
5768[2]	IBM 2-Port Gigabit Ethernet-SX PCI-e[7]	1000	Yes	Yes	Yes	No
573A[2]	IBM 10 Gigabit Ethernet-SX PCI-X	10000	Yes	No	Yes	No
181A[1]	IBM 2-Port 10/100/1000 Base-TX PCI-e[7]	10 / 100 / 1000	Yes	Yes	Yes	Yes
181B[2]	IBM 2-Port Gigabit Base-SX PCI-e	10000	Yes	Yes	Yes	Yes
181C[1]	IBM 4-Port 10/100/1000 Base-TX PCI-e[7]	10 / 100 / 1000	Yes	Yes	Yes	Yes
1819[1]	IBM 4-Port 10/100/1000 Base-TX PCI-e[7,9]	10 / 100 / 1000	Yes	Yes	Yes	Yes
N/A	Virtual Ethernet[4]	n/a[5]	Yes	N/A	Yes	No
N/A	Blade[8]	n/a[5]	Yes	N/A	Yes	Yes

Notes:

1. Unshielded Twisted Pair (UTP) card; uses copper wire cabling
2. Uses fiber optics
3. Custom Card Identification Number and System i Feature Code
4. Virtual Ethernet enables you to establish communication via TCP/IP between logical partitions and can be used without any additional hardware or software.
5. Depends on the hardware of the system.
6. These are theoretical hardware unidirectional speeds
7. Each port can handle 1000 Mbps
8. Blade communicates with the VIOS Partition via Virtual Ethernet
9. Host Ethernet Adapter for IBM Power 550, 9409-M50 running IBM i Operating System

- All adapters support Auto-negotiation

2.2 Communication Performance Test Environment

Hardware

All PCI-X measurements for 100 Mbps and 1 Gigabit were completed on an IBM System i 570+ 8-Way (2.2 GHz). Each system is configured as an LPAR, and each communication test was performed between two partitions on the same system with one dedicated CPU. The gigabit IOAs were installed in a 133MHz PCI-X slot.

The measurements for 10 Gigabit were completed on two IBM System i 520+ 2-Way (1.9 GHz) servers. Each System i server is configured as a single LPAR system with one dedicated CPU. Each communication test was performed between the two systems and the 10 Gigabit IOAs were installed in the 266 MHz PCI-X DDR(double data rate) slot for maximum performance. Only the 10 Gigabit Short Reach (573A) IOA's were used in our test environment.

All PCI-e measurements were completed on an IBM System i 9406-MMA 7061 16 way or IBM Power 550, 9409-M50. Each system is configured as an LPAR, and each communication test was performed between two partitions on the same system with one dedicated CPU. The Gigabit IOA's where installed in a PCI-e 8x slot.

All Blade Center measurements where collected on a 4 processor 7998-61X Blade in a Blade Center H chassis, 32 GB of memory. The AIX partition running the VIOS server was not limited. All performance data was collect with the Blade running as the server. The System i partition (on the Blade) was limited to 1 CPU with 4 GB of memory and communicated with an external IBM System i 570+ 8-Way (2.2 GHz) configured as a single LPAR system with one dedicated CPU and 4 GB of Memory.

Software

The NetPerf and Netop workloads are primitive-level function workloads used to explore communications performance. Workloads consist of programs that run between a System i client and a System i server, Multiple instances of the workloads can be executed over multiple connections to increase the system load. The programs communicate with each other using sockets or SSL APIs.

To demonstrate communications performance in various ways, several workload scenarios are analyzed. Each of these scenarios may be executed with regular nonsecure sockets or with secure SSL using the GSK API:

1. **Request/Response (RR):** The client and server send a specified amount of data back and forth over a connection that remains active.
2. **Asymmetric Connect/Request/Response (ACRR)**: The client establishes a connection with the server, a single small request (64 bytes) is sent to the server, and a response (8K bytes) is sent by the server back to the client, and the connection is closed.
3. **Large transfer (Stream):** The client repetitively sends a given amount of data to the server over a connection that remains active.

The NetPerf and Netop tools used to measure these benchmarks merely copy and transfer the data from memory. Therefore, additional consideration must be given to account for other normal application processing costs (for example, higher CPU utilization and higher response times due to disk access time). A real user application will have this type of processing as only a percentage of the overall workload. The IBM Systems Workload Estimator, described in Chapter 20, reflects the performance of real user applications while averaging the impact of the differences between the various communications protocols. The real world perspective offered by the Workload Estimator can be valuable for projecting overall system capacity.

2.3 Communication and Storage observations

With the continued progress in both communication and storage technology, it is possible that the performance bottleneck shifts. Especially with high bandwidth communication such as 10 Gigabit and Virtual ethernet, storage technology could become the limiting factor.

DASD Performance

Storage performance is dependent on the configuration and amount of disk units within your partition. See chapter 4 for detailed information.

Table 5.2

IOA and operation		Number of 35 GB DASD units (Measurement numbers in GB/HR)		
2778 IOA		15 Units	30 Units	45 Units
*SAVF	Save	41	83	122
	Restore	41	83	122
2757 IOA				
*SAVF	Save	82	165	250
	Restore	82	165	250

Large data transfer (FTP)

When transferring large amounts of data, for example with FTP, DASD performance plays an important role. Both the sending and receiving end could limit the communication speed when using high bandwidth communication. Also in a multi-threading environment, having more then one streaming session could improve overall communication performance when the DASD throughput is available.

Table 2.3

Virtual Ethernet	**Performance in MB per second**	
FTP	**1 Disk Unit ASP on 2757 IOA**	**15 Disk Units ASP on 2757 IOA**
1 Session	10.8	42.0
2 Sessions	10.5	70.0
3 Sessions	10.4	75.0

2.4 TCP/IP non-secure performance

In table 2.4 you will find the payload information for the different Ethernet types. The most important factor with streaming is to determine how much data can be transferred. The results are listed in bits and bytes per second. Virtual Ethernet does not have a raw bit rate, since the maximum throughput is determined by the CPU.

Table 2.4

Streaming Performance

Ethernet Type	**Raw bit rate[1] (Mbits per second)**	**MTU[2]**	**Payload Simplex[3] (Mbits per second)**	**Payload Duplex[4] (Mbits per second)**
100 Megabit	100	1,492	93.5	170.0
1 Gigabit	1,000	1,492	935.4	1740.3
		8,992	935.9	1753.1
10 Gigabit[5]	10,000	1,492	3745.4	4400.7
		8,992	8789.6	9297.0
HEA 1 Gigabit	1,000	1,492	986.4	1481.4
		8,992	941.1	1960.9
	160,00[7]	1,492	2811.8	6331.0
		8,992	9800.7	10586.4
HEA 10 Gigabit	10,000	1,492	2913.1	3305.2
		8,992	9392.3	9276.9
	160,00[7]	1,492	2823.5	6332.3
		8,992	9813.7	10602.3
Blade[8]	n/a	1,492	933.1	1014.4
Virtual[6]	n/a	8,992	8553.0	11972.3

Notes:

1. The Raw bit rate value is the physical media bit rate and does not reflect physical media overheads
2. Maximum Transmission Unit. The large (8992 bytes) MTU is also referred to as Jumbo Frames.
3. Simplex is a single direction TCP data stream.
4. Duplex is a bidirectional TCP data stream.
5. The 10 Gigabit results were obtained by using multiple sessions, because a single sessions is incapable to fully utilize the 10 Gigabit adapter.
6. Virtual Ethernet uses Jumbo Frames only, since large packets are supported throughout the whole connection path.
7. HEA P.P.U.T (Partition to Partition Unicast Traffic or internal switch) 16 Gbps per port group.
8. 4 Processor 7998-61X Blade
9. All measurements are performed with Full Duplex Ethernet.

Streaming data is not the only type of communication handled through Ethernet. Often server and client applications communicate with small packets of data back and forth (RR). In the case of web browsers, the most common type is to connect, request and receive data, then disconnect (ACRR). Table 2.5 provides some rough capacity planning information for these RR and ACRR communications.

Table 2.5

RR & ACRR Performance
(Transactions per second per server CPU)

Transaction Type	Threads	1 Gigabit	Virtual
Request/Response (RR) 128 Bytes	1	991.32	873.62
	26	1330.45	912.34
Asym. Connect/Request/Response (ACRR) 8K Bytes	1	261.51	218.82
	26	279.64	221.21

Notes:

- Capacity metrics are provided for nonsecure transactions
- The table data reflects System i as a server (not a client)
- The data reflects Sockets and TCP/IP
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- All measurement where taken with Packet Trainer off (See 2.6 for line dependent performance enhancements)

Here the results show the difference in performance for different Ethernet cards compared with Virtual Ethernet. We also added test results with multiple threads to give an insight on the performance when a system is stressed with multiple sessions.

This information is of similar type to that provided in Chapter 15, Web Server Performance. There are also capacity planning examples in that chapter.

2.5 TCP/IP Secure Performance

With the growth of communication over public network environments like the Internet, securing the communication data becomes a greater concern. Good examples are customers providing personal data to complete a purchase order (SSL) or someone working away from the office, but still able to connect to the company network (VPN).

SSL
SSL was created to provide a method of session security, authentication of a server or client, and message authentication. SSL is most commonly used to secure web communication, but SSL can be used for any reliable communication protocol (such as TCP). The successor to SSL is called TLS. There are slight differences between SSL v3.0 and TLS v1.0, but the protocol remains substantially the same. For the data gathered here we only use the TLS v1.0 protocol. Table 2.6 provides some rough capacity planning information for SSL communications, when using 1 Gigabit Ethernet.

Table 2.6

	SSL Performance (transactions per second per server CPU)					
Transaction Type:	**Nonsecure TCP/IP**	**RC4 / MD5**	**RC4 / SHA-1**	**AES128 / SHA-1**	**AES256 / SHA-1**	**TDES / SHA-1**
Request/Response (RR) 128 Byte	1167	565.4	530.0	479.6	462.1	202.2
Asym. Connect/Request/Response (ACRR) 8K Bytes	249.7	53.4	48.0	31.3	27.4	4.8
Large Transfer (Stream) 16K Bytes	478.4	55.7	53.3	36.9	31.9	6.5

Notes:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- Each SSL connection was established with a 1024 bit RSA handshake.

This table gives an overview on performance results on using different encryption methods in SSL compared to regular TCP/IP. The encryption methods we used range from fast but less secure (RC4 with MD5) to the slower but more secure (AES or TDES with SHA-1).

With SSL there is always a fixed overhead, such as the session handshake. The variable overhead is based on the number of bytes that need to be encrypted/decrypted, the size of the public key, the type of encryption, and the size of the symmetric key.

These results may be used to estimate a system's potential transaction rate at a given CPU utilization assuming a particular workload and security policy. Say the result of a given test is 5 transactions per second per server CPU. Then multiplying that result with 50 will tell that at 50% CPU utilization a transaction rate of 250 transactions per second is possible for this type of SSL communication on this environment. Similarly when a capacity of 100 transactions per second is required, the CPU utilization can be approximated by dividing 100 by 5, which gives a 20% CPU utilization in this environment. These are only estimations on how to size the workload, since actual results might vary. Similar information about SSL capacity planning can be found in Chapter 15, Web Server Performance.

Table 2.7 below illustrates relative CPU consumption for SSL instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 2.6. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

Table 2.7

	SSL Relative Performance (scaled to Nonsecure baseline)					
Transaction Type:	**Nonsecure TCP/IP**	**RC4 / MD5**	**RC4 / SHA-1**	**AES128 / SHA-1**	**AES256 / SHA-1**	**TDES / SHA-1**
Request/Response (RR) 128 Byte	1.0 x	2.1	2.2	2.4	2.5	5.8
Asym. Connect/Request/Response (ACRR) 8K Bytes	1.0 y	4.7	5.2	8.0	9.1	51.7
Large Transfer (Stream) 16K Bytes	1.0 z	8.6	9.0	13.0	15.0	73.7

Notes:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- Each SSL connections was established with a 1024 bit RSA handshake.
- x, y and z are scaling constants, one for each NetPerf scenario.

VPN

Although the term Virtual Private Networks (VPN) didn't start until early 1997, the concepts behind VPN started around the same time as the birth of the Internet. VPN creates a secure tunnel to communicate from one point to another using an unsecured network as media. Table 2.8 provides some rough capacity planning information for VPN communication, when using 1 Gigabit Ethernet.

Table 2.8

	VPN Performance (transactions per second per server CPU)				
Transaction Type:	**Nonsecure TCP/IP**	**AH with MD5**	**ESP with RC4 / MD5**	**ESP with AES128 / SHA-1**	**ESP with TDES / SHA-1**
Request/Response (RR) 128 Byte	1167.0	428.5	322.9	307.71	148.4
Asym. Connect/Request/Response (ACRR) 8K Bytes	249.7	49.9	37.7	32.7	9.1
Large Transfer (Stream) 16K Bytes	478.4	44.0	31.0	25.6	5.4

Notes:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- VPN measurements used transport mode, TDES, AES128 or RC4 with 128-bit key symmetric cipher and MD5 message digest with RSA public/private keys. VPN antireplay was disabled.
- This is only a rough indicator for capacity planning. Actual results may differ significantly.

This table also shows a range of encryption methods to give you an insight on the performance between less secure but faster, or more secure but slower methods, all compared to unsecured TCP/IP.

Table 2.9 below illustrates relative CPU consumption for VPN instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 2.6. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

Table 2.9

	VPN Relative Performance (scaled to Nonsecure baseline)				
Transaction Type:	**Nonsecure TCP/IP**	**AH with MD5**	**ESP with RC4 / MD5**	**ESP with AES128 / SHA-1**	**ESP with TDES / SHA-1**
Request/Response (RR) 128 Byte	1.0 x	2.7	3.6	3.8	7.9
Asym. Connect/Request/Response (ACRR) 8K Bytes	1.0 y	5.0	6.6	7.6	27.5
Large Transfer (Stream) 16K Bytes	1.0 z	10.9	15.4	18.7	88.8

Notes:

- Capacity metrics are provided for nonsecure and each variation of security policy
- The table data reflects System i as a server (not a client)
- VPN measurements used transport mode, TDES, AES128 or RC4 with 128-bit key symmetric cipher and MD5 message digest with RSA public/private keys. VPN anti-replay was disabled.
- This is only a rough indicator for capacity planning. Actual results may differ significantly.
- x, y and z are scaling constants, one for each NetPerf scenario.

The SSL and VPN measurements are based on a specific set of cipher methods and public key sizes. Other choices will perform differently.

2.6 Performance Observations and Tips

- Communication performance on Blades may see an increase when the processors are in shared mode. This is workload dependent.
- Host Ethernet Adapters require 40 to 56 MB for memory per logical port to vary on.
- IBM Power 550, 9409-M50 May show 2 to 5 percent increase over IBM Power 520, 9408-M25 due to the incorporation of L3 cache. Results will vary based on workload and configuration.
- Virtual ethernet should always be configured with jumbo frame enabled
- In 6.1 Packet Trainer is defaulted to "off" but can be configured per Line Description in 6.1.
- Virtual ethernet may see performance increases with Packet Trainer turn on. This depends on workload, connection type and utilization.
- Physical Gigabit lines may see performance increases with Packet Trainer off. This depends on workload, connection type and utilization.
- Host Ethernet Adapter should not be used for performance sensitive workloads, your throughput can be greatly affected by the use of other logical ports connected to your physical port on additional partitions.
- Host Ethernet Adapter may see performance increases with Packet Trainer set to on, especially with regard to HEA's internal Logical Switch and Partition to Partition traffic via the same port group.

- For additional information regarding your Host Ethernet Adapter please see your specification manual and the Performance Management page for future white papers regarding iSeries and HEA.

- 1 Gigabit Jumbo frame Ethernet enables 12% greater throughput compared to normal frame 1 Gigabit Ethernet. This may vary significantly based on your system, network and workload attributes. Measured 1 Gigabit Jumbo Frame Ethernet throughput approached 1 Gigabit/sec

- The jumbo frame option requires 8992 Byte MTU support by all of the network components including switches, routers and bridges. For System Adapter configuration, LINESPEED(*AUTO) and DUPLEX(*FULL) or DUPLEX(*AUTO) must also be specified. To confirm that jumbo frames have been successfully configured throughout the network, use NETSTAT option 3 to "Display Details" for the active jumbo frame network connection.

- Using ***ETHV2** for the "Ethernet Standard" attribute of **CRTLINETH** may see slight performance increase in STREAMING workloads for 1 Gigabit lines.

- Always ensure that the entire communications network is configured optimally. The **maximum frame size parameter** (MAXFRAME on LIND) should be maximized. The **maximum transmission unit (MTU) size** parameter (CFGTCP command) for both the interface and the route affect the actual size of the line flows and should be configured to *LIND and *IFC respectively. Having configured a large frame size does not negatively impact performance for small transfers. Note that both the System i and the other link station must be configured for large frames. Otherwise, the smaller of the two maximum frame size values is used in transferring data. Bridges may also limit the maximum frame size.

- When transferring large amounts of data, maximize the size of the application's send and receive requests. This is the amount of data that the application transfers with a single sockets API call. Because sockets does not block up multiple application sends, it is important to block in the application if possible.

- With the CHGTCPA command using the parameters TCPRCVBUF and TCPSNDBUF you can alter the TCP receive and send buffers. When transferring large amounts of data, you may experience higher throughput by increasing these buffer sizes up to 8MB. The exact buffer size that provides the best throughput will be dependent on several network environment factors including types of switches and systems, ACK timing, error rate and network topology. In our test environment we used 1 MB buffers. Read the help for this command for more information.

- Application time for transfer environments, including accessing a data base file, decreases the maximum potential data rate. Because the CPU has additional work to process, a smaller percentage of the CPU is available to handle the transfer of data. Also, serialization from the application's use of both database and communications will reduce the transfer rates.

- TCP/IP Attributes (CHGTCPA) now includes a parameter to set the TCP closed connection wait time-out value (TCPCLOTIMO) . This value indicates the amount of time, in seconds, for which a socket pair (client IP address and port, server IP address and port) cannot be reused after a connection is closed. Normally it is set to at least twice the maximum segment lifetime. For typical applications the default value of 120 seconds, limiting the system to approximately 500 new socket pairs per second, is fine. Some applications such as primitive communications benchmarks work best if this setting reflects a value closer to twice the true maximum segment lifetime. In these cases a setting of

only a few seconds may perform best. Setting this value too low may result in extra error handling impacting system capacity.

- No single station can or is expected to use the full bandwidth of the LAN media. It offers up to the media's rated speed of aggregate capacity for the attached stations to share. The disk access time is usually the limiting resource. The data rate is governed primarily by the application efficiency attributes (for example, amount of disk accesses, amount of CPU processing of data, application blocking factors, etc.).

- LAN can achieve a significantly higher data rate than most supported WAN protocols. This is due to the desirable combination of having a high media speed along with optimized protocol software.

- Communications applications consume CPU resource (to process data, to support disk I/O, etc.) and communications line resource (to send and receive data). The amount of line resource that is consumed is proportional to the total number of bytes sent or received on the line. Some additional CPU resource is consumed to process the communications software to support the individual sends (puts or writes) and receives (gets or reads).

- When several sessions use a line concurrently, the aggregate data rate may be higher. This is due to the inherent inefficiency of a single session in using the link. In other words, when a single job is executing disk operations or doing non-overlapped CPU processing, the communications link is idle. If several sessions transfer concurrently, then the jobs may be more interleaved and make better use of the communications link.

- The CPU usage for high speed connections is similar to "slower speed" lines running the same type of work. As the speed of a line increases from a traditional low speed to a high speed, performance characteristics may change.
    - Interactive transactions may be slightly faster
    - Large transfers may be significantly faster
    - A single job may be too serialized to utilize the entire bandwidth
    - High throughput is more sensitive to frame size
    - High throughput is more sensitive to application efficiency
    - System utilization from other work has more impact on throughput

- When developing scalable communication applications, consider taking advantage of the Asynchronous and Overlapped I/O Sockets interface. This interface provides methods for threaded client server model applications to perform highly concurrent and have memory efficient I/O. Additional implementation information is available in the Sockets Programming guide.

2.7 APPC, ICF, CPI-C, and Anynet

- Ensure that APPC is configured optimally for best performance: LANMAXOUT on the CTLD (for APPC environments): This parameter governs how often the sending system waits for an acknowledgment. Never allow LANACKFRQ on one system to have a greater value than LANMAXOUT on the other system. The parameter values of the sending system should match the values on the receiving system. In general, a value of *CALC (i.e., LANMAXOUT=2) offers the best performance for interactive environments, and adequate performance for large transfer environments. For large transfer environments, changing LANMAXOUT to 6 may provide a significant performance increase. LANWNWSTP for APPC on the controller description (CTLD): If

there is network congestion or overruns to certain target system adapters, then increasing the value from the default=*NONE to 2 or something larger may improve performance. MAXLENRU for APPC on the mode description (MODD): If a value of *CALC is selected for the maximum SNA request/response unit (RU) the system will select an efficient size that is compatible with the frame size (on the LIND) that you choose. The newer LAN IOPs support IOP assist. Changing the RU size to a value other than *CALC may negate this performance feature.

- Some APPC APIs provide blocking (e.g., ICF and CPI-C), therefore scenarios that include repetitive small puts (that may be blocked) may achieve much better performance.
- A large transfer with the System i sending each record repetitively using the default blocking provided by OS/400 to the System i client provides the best level of performance.
- A large transfer with the System i flushing the communications buffer after each record (FRCDTA keyword for ICF) to the System i client consumes more CPU time and reduces the potential data rate. That is, each record will be forced out of the server system to the client system without waiting to be blocked with any subsequent data. Note that ICF and CPI-C support blocking, Sockets does not.
- A large transfer with the System i sending each record requiring a synchronous confirm (e.g., CONFIRM keyword for ICF) to the System is client uses even more CPU and places a high level of serialization reducing the data rate. That is, each record is forced out of the server system to the client system. The server system program then waits for the client system to respond with a confirm (acknowledgment). The server application cannot send the next record until the confirm has been received.
- Compression with APPC should be used with caution and only for slower speed WAN environments. Many suggest that compression should be used with speeds 19.2 kbps and slower and is dependent on the data being transmitted (# of blanks, # and type of repetitions, etc.). Compression is very CPU-intensive. For the CPB benchmark, compression increases the CPU time by up to 9 times. RLE compression uses less CPU time than LZ9 compression (MODD parameters).
- ICF and CPI-C have very similar performance for small data transfers.
- ICF allows for locate mode which means one less move of the data. This makes a significant difference when using larger records.
- The best case data rate is to use the normal blocking that OS/400 provides. For best performance, the use of the ICF keywords force data and confirm should be minimized. An application's use of these keywords has its place, but the tradeoff with performance should be considered. Any deviation from using the normal blocking that OS/400 provides may cause additional trips through the communications software and hardware; therefore, it increases both the overall delay and the amount of resources consumed.
- Having ANYNET = *YES causes extra CPU processing. Only have it set to *YES if it is needed functionally; otherwise, leave it set to *NO.
- For send and receive pairs, the most efficient use of an interface is with it's "native" protocol stack. That is, ICF and CPI-C perform the best with APPC, and Sockets performs best with TCP/IP. There is CPU time overhead when the "cross over" is processed. Each interface/stack may perform differently depending on the scenario.
- Copyfile with DDM provides an efficient way to transfer files between System i systems. DDM provides large blocking which limits the number of times the communications support is invoked. It also maximizes efficiencies with the data base by doing fewer larger I/Os. Generally, a higher data rate can be achieved with DDM compared with user-written APPC programs (doing data base accesses) or with ODF.
- When ODF is used with the SNDNETF command, it must first copy the data to the distribution queue on the sending system. This activity is highly CPU-intensive and takes a considerable amount of time. This time is dependent on the number and size of the records in the file. Sending an object to more than one target System i server only requires one copy to the distribution queue. Therefore, the realized data rate may appear higher for the subsequent transfers.

- FTS is a less efficient way to transfer data. However, it offers built in data compression for line speeds less than a given threshold. In some configurations, it will compress data when using LAN; this significantly slows down LAN transfers.

2.8 HPR and Enterprise extender considerations

Enterprise Extender is a protocol that allows the transmission of APPC data over IP only infrastructure. In System i support for Enterprise Extender is added in 2.4. The communications using Enterprise Extender protocol can be achieved by creating a special kind of APPC controller, with LINKTYPE parameter of *HPRIP.

Enterprise Extender (*HPRIP) APPC controllers are not attached to a specific line. Because of this, the controller uses the LDLCLNKSPD parameter to determine the initial link speed to the remote system. After a connection has been started, this speed is adjusted automatically, using the measured network values. However if the value of LDLCLNKSPD is too different to the real link speed value at the beginning, the initial connections will not be using optimally the network. A high value will cause too many packets to be dropped, and a low value will cause the system not to reach the real link speed for short bursts of data.

In a laboratory controlled environment with an isolated 100 Mbps Ethernet network, the following average response times were observed on the system (**not** including the time required to start a SNA session and allocate a conversation):

Table 2.9

Test Type	**HPRIP Link Speed = 10Mbps**	**HPRIP Link Speed = 100Mbps**	**AnyNet**	**LAN**
Short Request with echo	0.001 sec	0.001 sec	0.001 sec	0.001 sec
Short Request	0.001 sec	0.001 sec	0.003 sec	0.003 sec
64K Request with echo	0.019 sec	0.010 sec	13 sec	2 sec
64K Request	0.019 sec	0.010 sec	5 sec	1 sec
1GB Request with echo	6:14 min	6:08 min	7:22 min	6:04 min
1GB Request	2:32 min	2:17 min	3:33 min	3:00 min
Send File using sndnetf (1GB)	5:12 min	5:16 min	5:40 min	5:23 min

The tests were done between two IBM System i5 (9406-820 and 9402-400) servers in an isolated network.

Allocation time refers to the time that it takes for the system to start a conversation to the remote system. The allocation time might be greater when a SNA session has not yet started to the remote system. Measured allocation speed times where of 14 ms, in HPRIP systems in average, while in AnyNet allocation times where of 41 ms in average.

The HPRIP controllers have slightly higher CPU usage than controllers that use a direct LAN attach. The CPU usage is similar to the one measured on AnyNet APPC controllers. On laboratory testing, a LAN transaction took 3 CPW, while HPRIP and AnyNet, both took 3.7 CPW**.**

2.9 Additional Information

Extensive information can be found at the System i Information Center web site at: http://www.ibm.com/eserver/iseries/infocenter .

- For network information select *"Networking"*:
  - See *"TCP/IP setup" → "Internet Protocol version 6"* for IPv6 information
  - See *"Network communications" → "Ethernet"* for Ethernet information.
- For application development select *"Programming"*:
  - See *"Communications" → "Socket Programming"* for the Sockets Programming guide.

Information about Ethernet cards can be found at the IBM Systems Hardware Information Center. The link for this information center is located on the IBM Systems Information Centers Page at: http://publib.boulder.ibm.com/eserver .

- See *"Managing your server and devices" → "Managing devices" → "Managing Peripheral Component Interconnect (PCI) adapters"* for Ethernet PCI adapters information.

Chapter 3. Cryptography Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

With an increasing demand for security in today's information society, cryptography enables us to encrypt the communication and storage of secret or confidential data. This also requires data integrity, authentication and transaction non-repudiation. Together, cryptographic algorithms, shared/symmetric keys and public/private keys provide the mechanisms to support all of these requirements. This chapter focuses on the way that System i cryptographic solutions improve the performance of secure e-Business transactions.

There are many factors that affect System i performance in a cryptographic environment. This chapter discusses some of the common factors and offers guidance on how to achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and other performance workloads. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics.

Cryptography Performance Highlights for i5/OS V5R4M0:

- Support for the 4764 Cryptographic Coprocessor is added. This adapter provides both cryptographic coprocessor and secure-key cryptographic accelerator function in a single PCI-X card.
- 5722-AC3 Cryptographic Access Provider withdrawn. This product is no longer required to enable data encryption.
- Cryptographic Services API function added. Key management function has been added, which helps you securely store and handle cryptographic keys.

3.1 System i Cryptographic Solutions

On a System i, cryptographic solutions are based on software and hardware Cryptographic Service Providers (CSP). These solutions include services required for Network Authentication Service, SSL/TLS, VPN/IPSec, LDAP and SQL.

IBM Software Solutions

The software solutions are either part of the i5/OS Licensed Internal Code or the Java Cryptography Extension (JCE).

IBM Hardware Solutions

One of the hardware based cryptographic offload solutions for the System i is the **IBM 4764 PCI-X Cryptography Coprocessor (Feature Code 4806).** This solution will offload portions of cryptographic processing from the host CPU. The host CPU issues requests to the coprocessor hardware. The hardware then executes the cryptographic function and returns the results to the host CPU. Because this hardware based solution handles selected compute-intensive functions, the host CPU is available to support other

system activity. SSL/TLS network communications can use these options to dramatically offload cryptographic processing related to establishing an SSL/TLS session.

CSP API Sets

User applications can utilize cryptographic services indirectly via i5/OS functions (SSL/TLS, VPN IPSec) or directly via the following APIs:

- The Common Cryptographic Architecture (CCA) API set is provided for running cryptographic operations on a Cryptographic Coprocessor.
- The i5/OS Cryptographic Services API set is provided for running cryptographic operations within the Licensed Internal Code.
- Java Cryptography Extension (JCE) is a standard extension to the Java Software Development Kit (JDK).
- GSS (Generic Security Services), Java GSS, and Kerberos APIs are part of the Network Authentication Service that provides authentication and security services. These services include session level encryption capability.
- i5/OS SSL and JSSE support the Secure Sockets Layer Protocol. APIs provide session level encryption capability.
- Structured Query Language is used to access or modify information in a database. SQL supports encryption/decryption of database fields.

3.2 Cryptography Performance Test Environment

All measurements were completed on an IBM System i5 570+ 8-Way (2.2 GHz). The system is configured as an LPAR, and each test was performed on a single partition with one dedicated CPU. The partition was solely dedicated to run each test. The IBM 4764 PCI-X Cryptographic Coprocessor card is installed in a PCI-X slot.

This System i model is a POWER5 hardware system, which provides Simultaneous Multi-Threading. The tools used to obtain this data are in some cases only single threaded (single instruction stream) applications, which don't take advantage of the performance benefits of SMT. See section 3.6 for additional information.

Cryptperf is an IBM internal use primative-level cryptographic function test driver used to explore and measure System i cryptographic performance. It supports parameterized calls to various i5/OS CSPs. See section 3.6 for additional information.

- **Cipher:** Measures the performance of either symmetric or asymmetric key encrypt depending on algorithm selected.
- **Digest:** Measures the performance of hash functions.
- **Sign:** Measures the performance of hash with private key encrypt .
- **Pin:** Measures encrypted PIN verify using the IBM 3624 PIN format with the IBM 3624 PIN calculation method.

All i5/OS and JCE test cases run at a near 100% CPU utilization. The test cases that use the Cryptographic Coprocessor will offload all cryptographic functions, so that CPU utilization is negligible.

The relative performance and recommendations found in this chapter are similar for other models, but the data presented here is not representative of a specific customer environment. Cryptographic functions are very CPU intensive and scale easily. Adding or removing CPU's to an environment will change performance, so results in other environments may vary significantly.

3.3 Software Cryptographic API Performance

This section provides performance information for System i systems using the following cryptographic services; i5/OS Cryptographic Services API and IBM JCE 1.2.1, an extension of JDK 1.4.2.

Cryptographic performance is an important aspect of capacity planning, particularly for applications using secure network communications. The information in this section may be used to assist in capacity planning for this complex environment.

Measurement Results

The cryptographic performance measurements in the following three tables were made using i5/OS Cryptographic Services API and Java Cryptography Extension.

Table 3.1

Cipher Encrypt Performance

Encryption Algorithm	Threads	Key Length (Bits)	Transaction Length (Bytes)	i5/OS (Transactions/ Second)	i5/OS (Bytes/ Second)	JCE (Transactions /Second)	JCE (Bytes/ Second)
DES	1	56	1024	11,276	11,547,058	15,537	15,909,515
DES	10	56	1024	15,402	15,771,656	19,768	20,241,955
Triple DES	1	112	1024	5,039	5,159,756	5,997	6,140,893
Triple DES	1	112	65536	87	5,710,925	93	6,086,464
Triple DES	10	112	1024	6,625	6,783,658	7,517	7,697,917
Triple DES	10	112	65536	109	7,139,814	117	7,657,551
RC4	1	128	262144	947	248,224,207	125	32,704,635
RC4	10	128	262144	1,017	266,579,889	207	54,321,919
AES	1	128	1024	26,636	27,275,585	28,110	28,784,259
AES	1	128	65536	1,479	96,930,853	428	28,080,038
AES	1	256	1024	24,025	24,601,428	22,767	23,313,526
AES	1	256	65536	1,111	72,782,397	345	22,614,607
AES	10	128	1024	30,408	31,137,523	34,916	35,754,190
AES	10	128	65536	1,692	110,892,831	524	34,350,709
AES	10	256	1024	27,349	28,005,446	27,172	27,824,575
AES	10	256	65536	1,257	82,392,038	415	27,183,773
RSA	1	1024	100	897	n/a	197	n/a
RSA	1	2048	100	128	n/a	30	n/a
RSA	10	1024	100	1,187	n/a	246	n/a
RSA	10	2048	100	165	n/a	35	n/a

Notes:
- See section 3.2 for Test Environment Information

Table 3.2

Signing Performance

Encryption Algorithm	Threads	RSA Key Length (Bits)	i5/OS (Transactions/Second)	JCE (Transactions/Second)
SHA-1 / RSA	1	1024	901	197
SHA-1 / RSA	10	1024	1,155	240
SHA-1 / RSA	1	2048	129	30
SHA-1 / RSA	10	2048	163	35

Notes:
- Transaction Length set at 1024 bytes
- See section 3.2 for Test Environment Information

Table 3.3

Digest Performance

Encryption Algorithm	Threads	i5/OS (Transactions/ Second)	i5/OS (Bytes/ Second)	JCE (Transactions/ Second)	JCE (Bytes/Second)
SHA-1	1	6,753	110,642,896	2,295	37,608,172
SHA-1	10	10,875	178,172,751	2,954	48,401,773
SHA-256	1	3,885	63,645,228	2,049	33,576,523
SHA-256	10	4,461	73,086,411	2,392	39,184,923
SHA-384	1	7,050	115,505,548	4,020	65,865,327
SHA-384	10	8,075	132,301,878	4,634	75,925,668
SHA-512	1	7,031	115,201,800	4,217	69,098,731
SHA-512	10	8,060	132,059,807	4,801	78,659,561

Notes:
- Key Length set at 1024 bits
- Transaction Length set at 16384 bytes
- See section 3.2 for Test Environment Information

3.4 Hardware Cryptographic API Performance

This section provides information on the hardware based cryptographic offload solution **IBM 4764 PCI-X Cryptography Coprocessor (Feature Code 4806)**. This solution will improve the system CPU capacity by offloading CPU demanding cryptographic functions.

IBM Common Name	**IBM 4764 PCI-X Cryptographic Coprocessor**
System i hardware feature code	#4806
Applications	Banking/finance (B/F)
	Secure accelerator (SSL)
Cryptographic Key Protection	Secure hardware module
Required Hardware	No IOP Required
Platform Support	IBM System i5

The 4764 Cryptographic Coprocessor provides both cryptographic coprocessor and secure-key cryptographic accelerator functions in a single PCI-X card. The coprocessor functions are targeted to banking and finance applications. The secure-key accelerator functions are targeted to improving the performance of SSL (secure socket layer) and TLS (transport layer security) based transactions. The 4764 Cryptographic Coprocessor supports secure storage of cryptographic keys in a tamper-resistant module,

which is designed to meet FIPS 140-2 Level 4 security requirements. This new cryptographic card offers the security and performance required to support e-Business and emerging digital signature applications.

For banking and finance applications the 4764 Cryptographic Coprocessor delivers improved performance for T-DES, RSA, and financial PIN processing. IBM CCA (Common Cryptographic Architecture) APIs are provided to enable finance and other specialized applications to access the services of the coprocessor. For banking and finance applications the 4764 Coprocessor is a replacement for the 4758-023 Cryptographic Coprocessor (feature code 4801).

The 4764 Cryptographic Coprocessor can also be used to improve the performance of high-transaction-rate secure applications that use the SSL and TLS protocols. These protocols are used between server and client applications over a public network like the Internet, when private information is being transmitted in the case of Consumer-to-Business transactions (for example, a web transaction with payment information containing credit card numbers) or Business-to-Business transactions. SSL/TLS is the predominant method for securing web transactions. Establishing SSL/TLS secure web connections requires very compute intensive cryptographic processing. The 4764 Cryptographic Coprocessor off-loads cryptographic RSA processing associated with the establishment of a SSL/TLS session, thus freeing the server for other processing. For cryptographic accelerator applications the 4764 Cryptographic Coprocessor is a replacement for the 2058 Cryptographic Accelerator (feature code 4805).

Cryptographic performance is an important aspect of capacity planning, particularly for applications using SSL/TLS network communications. Besides host processing capacity, the impact of one or more Cryptographic Coprocessors must be considered. Adding a Cryptographic Coprocessor to your environment can often be more beneficial then adding a CPU. The information in this chapter may be used to assist in capacity planning for this complex environment.

Measurement Results

The following three tables display the cryptographic test cases that use the Common Cryptographic Architecture (CCA) interface to measure transactions per second for a variety of 4764 Cryptographic Coprocessor functions.

Table 3.4

Cipher Encrypt Performance
CCA CSP

Encryption Algorithm	Threads	Key Length (Bits)	Transaction Length (Bytes)	4764 (Transactions/second)	4764 (Bytes/second)
DES	1	56	1024	1,026	1,050,283
DES	10	56	1024	1,053	1,078,458
Triple DES	1	112	1024	1,002	1,025,798
Triple DES	1	112	65536	110	7,191,327
Triple DES	10	112	1024	1,021	1,045,535
Triple DES	10	112	65536	123	8,035,164
RSA	1	1024	100	796	n/a
RSA	1	2048	100	307	n/a
RSA	10	1024	100	1,044	n/a
RSA	10	2048	100	462	n/a

Notes:
- See section 3.2 for Test Environment information
- AES is not supported by the IBM 4764 Cryptographic Coprocessor

Table 3.5

Signing Performance CCA CSP			
Encryption Algorithm	**Threads**	**RSA Key Length (Bits)**	**4764 (Transactions/second)**
SHA-1 / RSA	1	1024	794
SHA-1 / RSA	10	1024	1,074
SHA-1 / RSA	1	2048	308
SHA-1 / RSA	10	2048	465

Notes:
- Transaction Length set at 1024 bytes
- See section 3.2 for Test Environment information

Table 3.6

Financial PINs Performance CCA CSP		
Threads	**Total Repetitions**	**4764 (Transactions/second)**
1	10000	945
10	100000	966

Notes:
- See section 3.2 for Test Environment information

3.5 Cryptography Observations, Tips and Recommendations

- The IBM Systems Workload Estimator, described in Chapter 20, reflects the performance of real user applications while averaging the impact of the differences between the various communications protocols. The real world perspective offered by the Workload Estimator may be valuable in some cases

- SSL/TLS client authentication requested by the server is quite expensive in terms of CPU and should be requested only when needed. Client authentication full handshakes use two to three times the CPU resource of server-only authentication. RSA authentication requests can be offloaded to an IBM 4764 Cryptographic Coprocessor.

- With the use of Collection Services you can count the SSL/TLS handshake operations. This capability allows you to better understand the performance impact of secure communications traffic. Use this tool to count how many full versus cached handshakes per second are being serviced by the server. Start the Collection Services with the default “Standard plus protocol”. When the collection is done you can find the SSL/TLS information in the QAPMJOBMI database file in the fields JBASH (full) and JBFSHA (cached) for server authentications or JBFSHA (full) and JBASHA (cached) for server and client authentications. Accumulate the full handshake numbers for all jobs and you will have a good method to determine the need for a 4764 Cryptographic Coprocessor. Information about Collection Services can be found at the System i Information Center. See section 3.6 for additional information.

- Symmetric key encryption and signing performance improves significantly when multithreaded.

- Supported number of 4764 Cryptographic Coprocessors:

Table 3.8

server models	Maximum per server	Maximum per partition
IBM System i5 570 8/12/16W, 595	32	8

IBM System i5 520, 550, 570 2/4W	8	8

- Applications requiring a FIPS 140-2 Level 4 certified, tamper resistant module for storing cryptographic keys should use the IBM 4764 Cryptographic Coprocessor.

- Cryptographic functions demand a lot of a system CPU, but the performance does scale well when you add a CPU to your system. If your CPU handles a large number of cryptographic requests, offloading them to an IBM 4764 Cryptographic Coprocessor might be more beneficial then adding a new CPU.

3.6 Additional Information

Extensive information about using System i Cryptographic functions may be found under “Security” and “Networking Security” at the System i Information Center web site at: http://www.ibm.com/eserver/iseries/infocenter .

IBM Security and Privacy specialists work with customers to assess, plan, design, implement and manage a security-rich environment for your online applications and transactions. These Security, Privacy, Wireless Security and PKI services are intended to help customers build trusted electronic relationships with employees, customers and business partners. These general IBM security services are described at: http://www.ibm.com/services/security/index.html .

General security news and information are available at: http://www.ibm.com/security .

System i Security White Paper, “Security is fundamental to the success of doing e-business” is available at: http://www.ibm.com/security/library/wp_secfund.shtml .

IBM Global Services provides a variety of Security Services for customers and Business Partners. Their services are described at: http://www.ibm.com/services/ .

Links to other Cryptographic Coprocessor documents including custom programming information can be found at: http://www.ibm.com/security/cryptocards .

Other performance information can be found at the System i Performance Management website at: http://www.ibm.com/systems/power/software/i/management/performance/index.html

Chapter 4. Internal Storage Performance

This chapter discusses DASD subsystems available for the System i platform.

There are two separate considerations. Before IBM i operating system V6R1, one only had to consider particular devices, IOAs, IOPs, and SAN devices. All attached through similar strategies directly to IBM i operating system and were all supported natively.

Starting in IBM iV6R1, however, IBM i operating system will be permitted to become a virtual client of an IBM product known as VIOS. The supported BladeCenter Blades will only be available in this fashion. For other IBM Power Systems it will be possible to attach all or some of the disks in this manner. See the appropriate chapters for information on SAN, VIOS and Blades. For previous IOAs shown here see previous performance capabilities references.

4.1 Internal (Native) Attachment

This section is intended to show relative performance differences in Disk Controllers which we will refer to as IOAs, and DASD, for customers to compare some of the available hardware. The workload used for our throughput measurements should not be used to gauge the workload capabilities of your system, since it is not a customer like workload.

The workload is designed to concentrate more on DASD, IOAs and IOPs, not the system as a whole. Workload throughput is not a measurement of operations per second but an activity counter in the workload itself. No LPAR's were used, all system resources were dedicated to the testing. The workload is batch and I/O intensive (small block reads and writes).

This chapter refers to disk drives and disk controllers (IOAs) using their CCIN number/code. The CCIN is what the system uses to understand what components are installed and is unique by each device. It is a four character, alphanumeric code. When you use commands in IBM i operating system to print your system configuration like PRTSYSINF or use the WRKHDWRSC *STG command to display hardware configuration information for your storage devices like the 571E or 571F disk controllers you see a listing of CCIN codes.

Note that the feature codes used in IBM's ordering system, e-config tool and inventory records are a four character numeric code which may or may not match the CCIN. IBM will sometimes use different features for the exact same physical device in order to communicate how the hardware is configured to the e-config tool or to provide packaging or pricing structures for the disk drive or IOA. For example, feature code 5738 and 5777 both identify a 571E IOA. A fairly complete list of CCIN and their feature codes can be found
POWER7 : http://publib.boulder.ibm.com/infocenter/powersys/v3r1m5/index.jsp?topic=/p7hcd/pcibyfeature.htm

4.1.0 Hardware Characteristics

Devices, Controllers & Enclosures

Technology	CCIN Codes	Approximate Size (GB)	RPM	Seek Time (ms)		Latency (ms)	Max Interface Speed [1] (MB/s)
				Read	Write		
SCSI	4326	35	15K	3.6	4.0	2	320
	4327	70	15K	3.6	4.0	2	320
	4328	139	15K	3.6	4.0	2	320
	4329	283	15K	3.6	4.0	2	320
SAS	433B	70	15K	3.5	4.0	2	300
	433C	139	15K	3.5	4.0	2	300
	433D	280	15K	3.5	4.0	2	300
	198B	70	15K	3.0	3.5	2	300
	198C	139	15K	3.1	3.5	2	300
	19B0	139	15K	3.1	3.5	2	600
	198C	139	15K	3.1	3.5	2	300
	19A1	283	15K	2.9	3.3	2	300
	19B1	283	15K	2.9	3.3	2	600
	198E	428	15K	3.6	4.1	2	300
	198D	283	10K	3.7	4.2	3	300
	19B7	283	10K	3.7	4.2	3	600
	19A3	571	10K	3.7	4.2	3	300
	19B3	571	10K	3.7	4.2	3	600
SSD	58B0	70	N/A	0	0	0	300
	58B2	177	N/A	0	0	0	300
	58B3	177	N/A	0	0	0	300
	58B4	177	N/A	0	0	0	300
	58B8	387	N/A	0	0	0	600
	58B9	387	N/A	0	0	0	600

	Enclosure CCIN's (or Feature Code if no CCIN exists)	Max # of devices [2]	Max Interface Speed [1] (MB/s)
	5095, 0595	12	160
	5094/5294	45/90	160
	28D2, 28DF, 28F5, 28F6, 292C, 292D, 292E	4	320
	28DB, 28B9, FC 7868	6	320
	FC 5786, FC 5787	24	320
	FC 5797, FC 5798	16	320
	28A8, 28A6, 2893, 2875, FC 5668,	6	300
	2876	8	300
	FC 5802	18	300
	FC 5803	26	300
	FC 5886	12	300
	FC 5887	24	600

[1] The actual drive interface speed (MB/s) is the minimum value of the speeds of the drive, the enclosure and the IOA. [2] Not all disk enclosures support the maximum number of disks in a RAID set.

CCIN Codes	(IOA) Feature Codes	Cache Non-compressed / up to compressed	Devices per RAID set [2]	Max Interface Speed [1] (MB/s)
2780	5580, 2780, 5590	235 MB write/up to 757 256 MB read/up to 1GB	3/18	320
573D	5727, 5728, 9510	40 MB	3/8	NA
57B8/57B7	5679	175 MB	3/20 RAID5 4/20 RAID6	300
571E/574F	5738, 5777, 5582, 5583	390 MB write/up to 1.5GB 415 MB read/up to 1.6GB	3/18 RAID5 4/18 RAID6	320
571F/575B	5739, 5778, 5781, 5782, 5799, 5800	390 MB write/up to 1.5GB 415 MB read/up to 1.6 GB	3/18 RAID5 4/18 RAID6	320
572C		NA	NA	300
572A	5900 , 0664, 5906, 5779, 5912	NA	NA	300
572F/575C	5904, 5906, 5908	390 MB write/up to 1.5GB 415 MB read/up to 1.6GB	3/18 RAID5 4/18 RAID6	300
574E	5903 5805	380 MB	3/18 RAID5 4/18 RAID6	300
57CF	5652 5662	175 MB	3/18 RAID5 4/18 RAID6	300
57CD	2053, 2054, 2055	0	4/4RAID5 4/4 RAID6	300
2BE1/2BD9	5631	175 MB	3/20 RAID5 4/20 RAID6	300
57B5	5913	1.5 GB	3/32 RAID5 4/32 RAID6	600
57C4	ESA1 ESA2	0	3/32 RAID5 4/32 RAID6	600

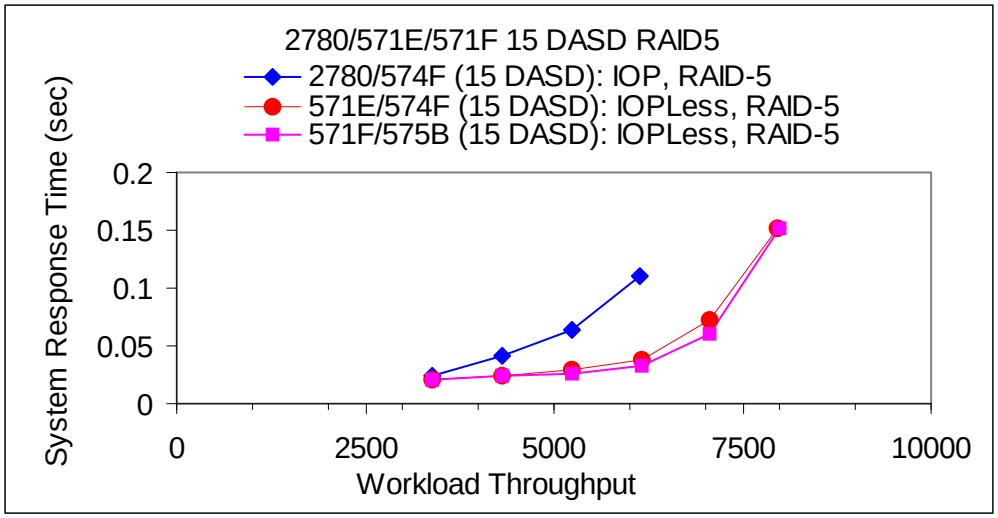
[1] The actual drive interface speed (MB/s) is the minimum value of the speeds of the drive, the enclosure and the IOA. [2] Not all disk enclosures support the maximum number of disks in a RAID set.

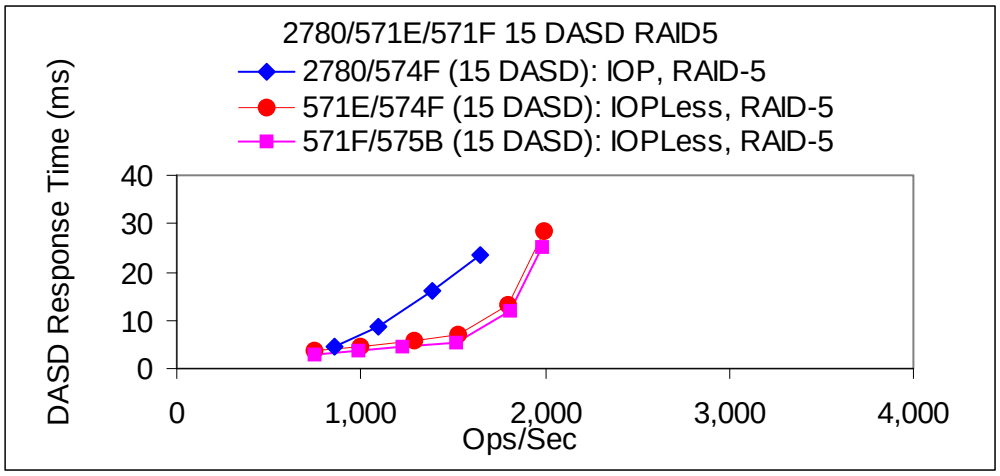
4.1.1 Comparing Current 2780/574F with the new 571E/574F and 571F/575B

NOTE: iV5R3 has support for the features in this section but all of our performance measurements were done on iV5R4 systems. For information on the supported features see the IBM Product Announcement Letters.

A model 570 4 way system with 48 GB of mainstore memory was used for the following. In comparing the new 571E/574F and 571F/575B with the current 2780/574F IOA, the larger read and write cache available on the new IOA's can have a very positive effect on workloads. Remember this workload is used to get a general comparison between new and current hardware and cannot predict what will happen with all workloads.

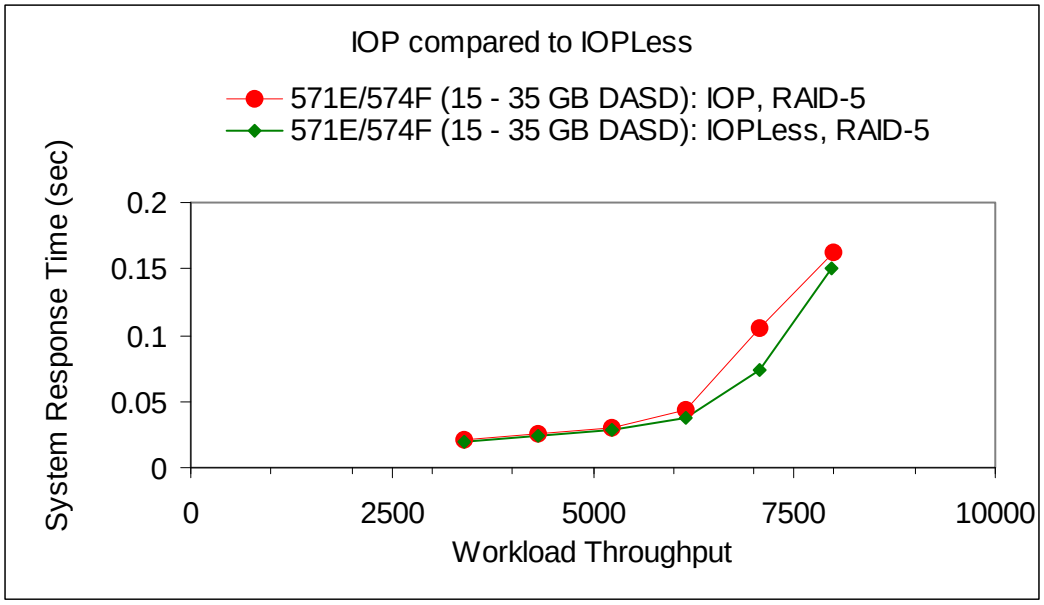
Also note the 571E/574F requires the auxiliary cache card to turn on RAID and the 571F/575B has the function included in its double-wide card packaging for better system protection. Understanding of the general results are intended to help customers gauge what might happen in their environments.

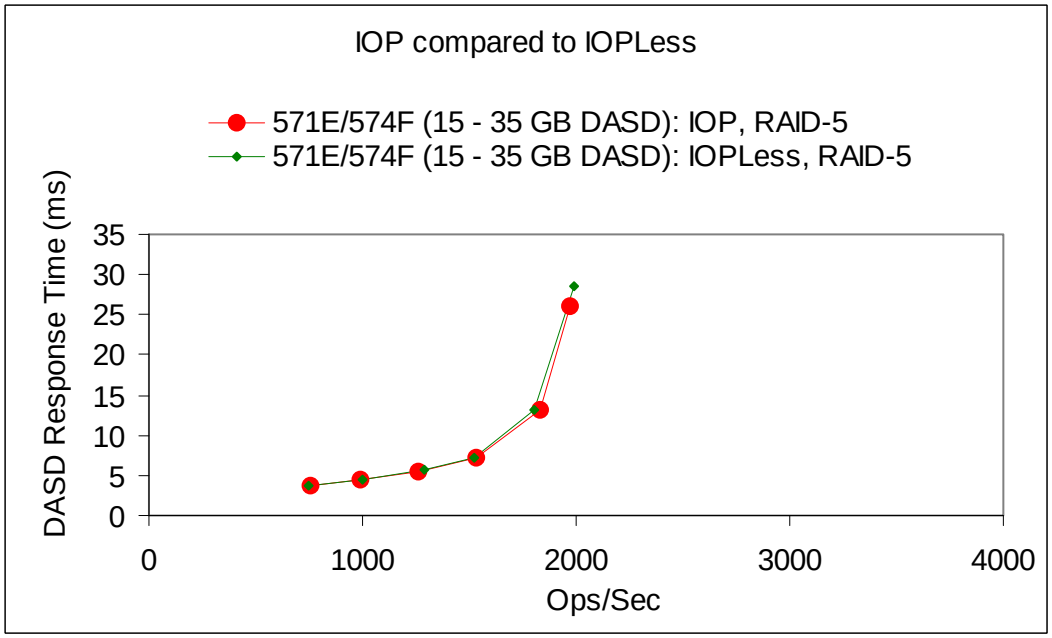




4.1.2 Comparing 571E/574F and 571F/575B IOP and IOPLess

In comparing IOP and IOPLess runs we did not see any significant differences, including the system CPU used. The system we used was a model 570 4 way, on the IOP run the system CPU was 11.6% and on the IOPLess run the system CPU was 11.5%. The 571E/574F and 571F/575B display similar characteristics when comparing IOP and IOPLess environments, so we have chosen to display results from only the 571E/574F.





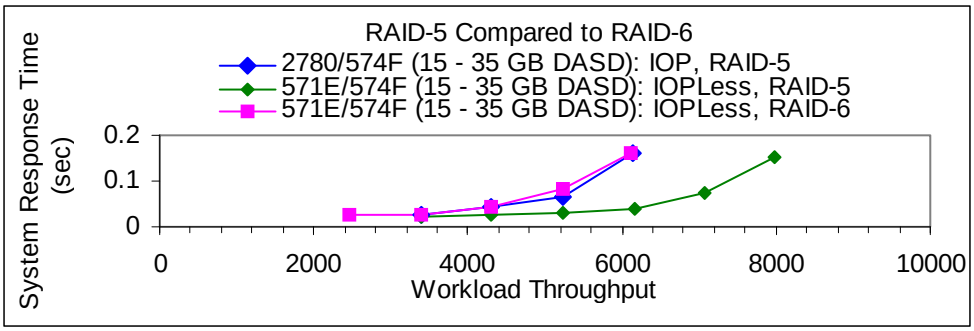
4.1.3 Comparing 571E/574F and 571F/575B RAID5 and RAID6 and Mirroring

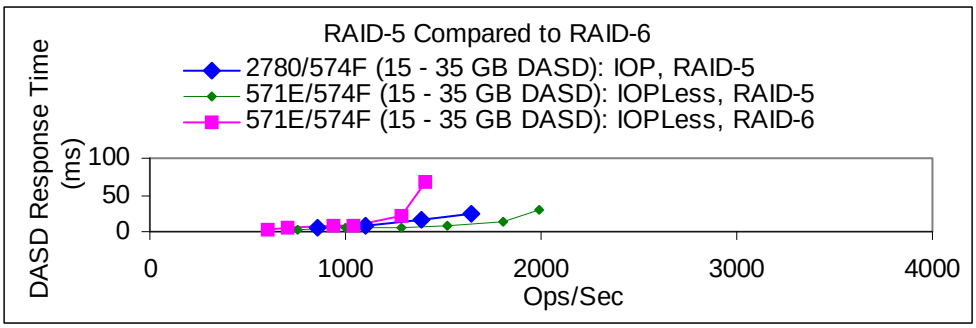
System i protection information can be found at http://www.redbooks.ibm.com/ in the current System i Handbook or the Info Center http://publib.boulder.ibm.com/iseries/ . When comparing RAID5, RAID6 and Mirroring we are interested in looking at the strength of failure protection vs storage capacity vs the performance impacts to the system workloads.

A model 570 4 way system with 48 GB of mainstore memory was used for the following. First comparing characteristics of RAID5 and RAID6; a customer can use Operations Navigator to better control the number of DASD in a RAID set but for this testing we signed on at DST and used default available to turn on our protection schemes. When turning on RAID5 the system configured two RAID sets under our IOA, one with 9 DASD and one with 6 DASD with a total disk capacity of 457 GB. For RAID6 the system created one RAID set with 15 DASD and a capacity of 456 GB. This would generally be true for most customer configurations.

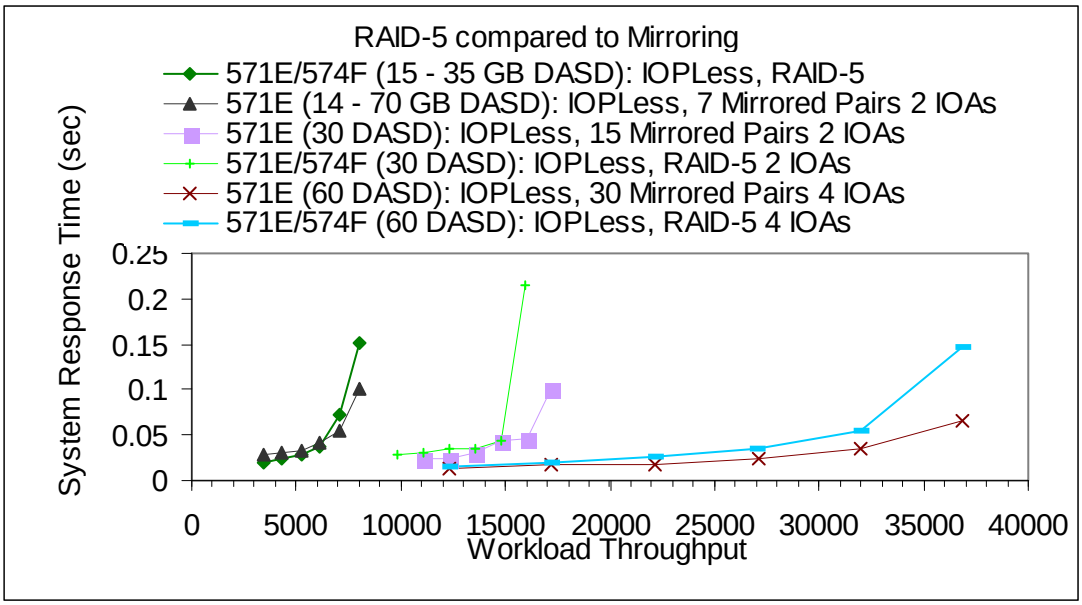
As you look at our run information you will notice that the performance boundaries of RAID6 on the 571E/574F is about the same as the performance boundaries of our 2780/574F configured using RAID5, so better protection could be achieved at current performance levels.

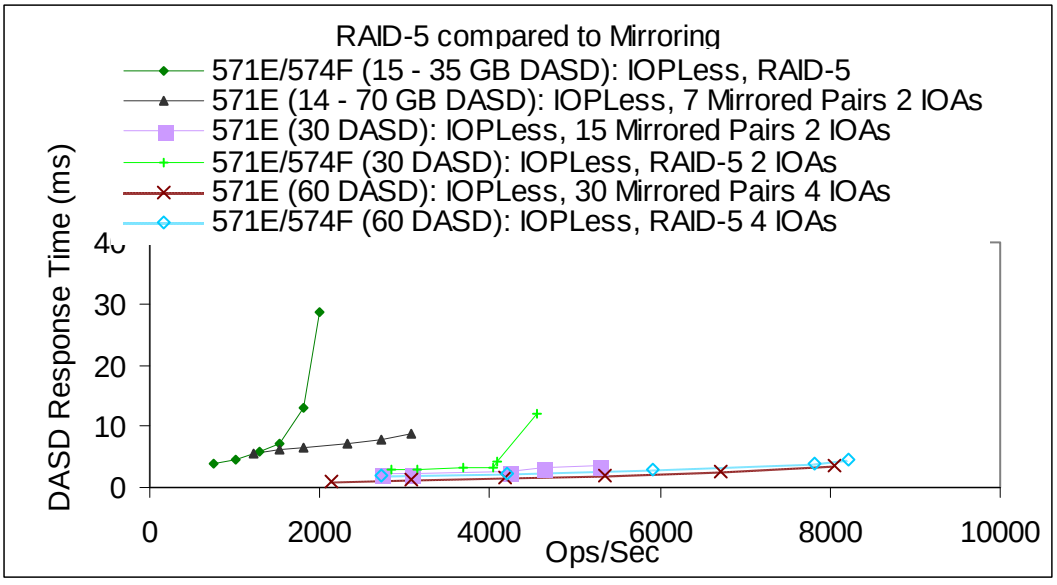
Another point of interest is that as long as a system is not pushing the boundaries, performance is similar in both the RAID5 and RAID6 environments. RAID6 is overwhelmed quicker than RAID5, so if RAID6 is desired for protection and the system workloads are approaching the boundaries, DASD and IOAs may need to be added to the system to achieve the desired performance levels. NOTE: If customers need better protection greater than RAID5 it might be worth considering the IOA level mirroring information on the following page.





In comparing Mirroring and RAID one of the concerns is capacity differences and the hardware needed. We tried to create an environment where the capacity was the same in both environments. To do this we built the same size database on "15 35GB DASD using RAID5" and "14 70GB DASD using Mirroring spread across 2 IOAs". The protection in the Mirrored environment is better but it also has the cost of an extra IOA in this low number DASD environment. For the 30 DASD and 60 DASD environments the number of IOAs needed is equal.

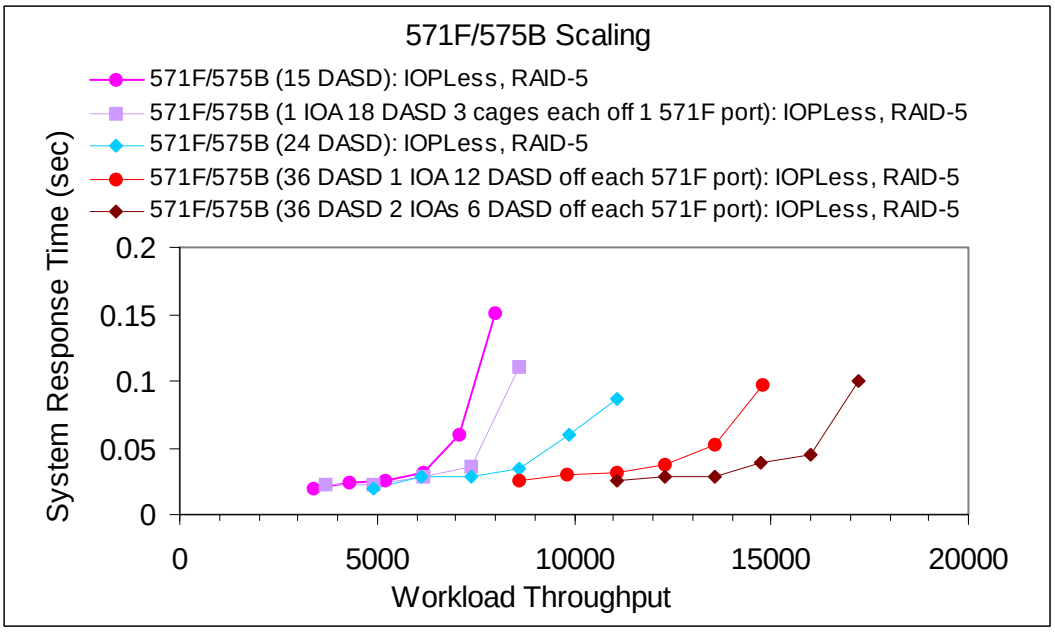


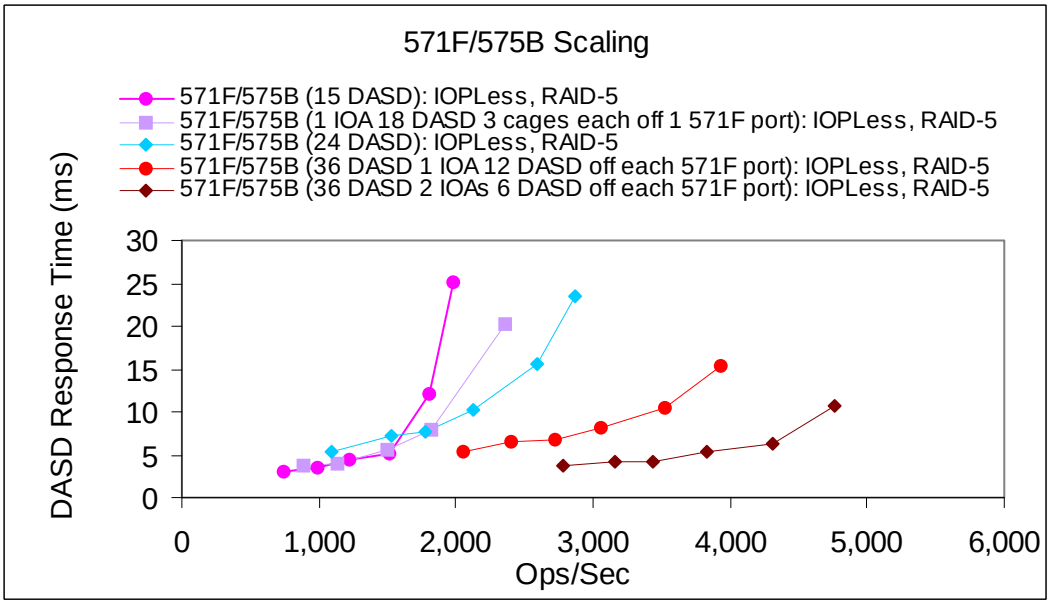


4.1.4 Performance Limits on the 571F/575B

In the following charts we try to characterize the 571F/575B in different DASD configuration. The 15 DASD experiment is used to give a comparison point with DASD experiments from chart 4.1.5.1 and 4.1.5.2. The 18, 24 and 36 DASD configurations are used to help in the discussion of performance vs capacity.

Our DASD IO workload scaled well from 15 DASD to 36 DASD on a single 571F/575B





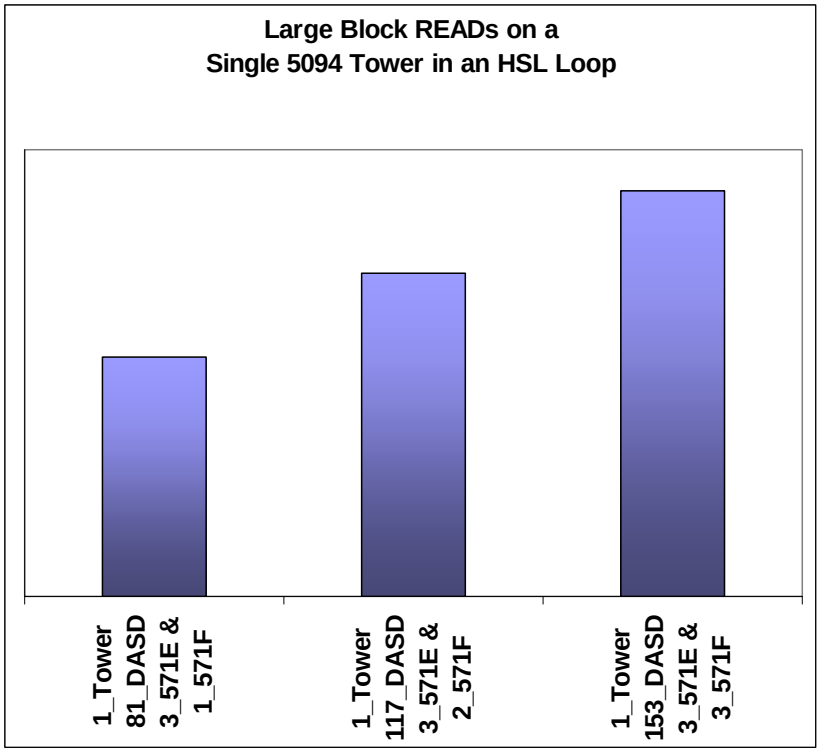
4.1.5 Investigating 571E/574F and 571F/575B IOA, Bus and HSL limitations

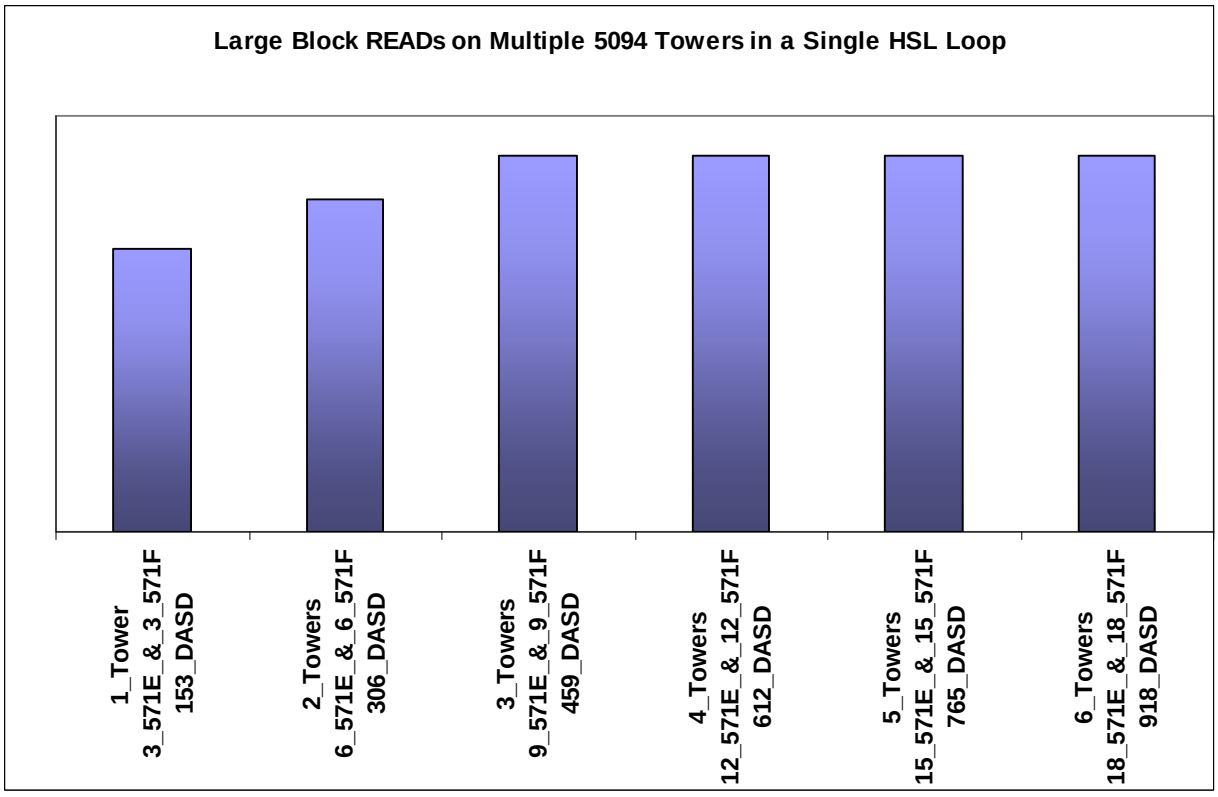
With the new DASD controllers and IOPLess capabilities, IBM has created many new options for our customers. Customers who needed more storage in their smaller configurations can now grow. With the ability to add more storage into an HSL loop the capacity and performance have the potential to grow. In the past a single HSL loop only allowed 6 5094 towers with 45 DASD per tower, giving a loop a capacity of 270 DASD, with the new DASD controllers that capacity has grown to 918 DASD. With the new configurations, you can see that 500 and even 600 DASD could make better use of the HSL loop's potential as opposed to the current limit of 270 DASD. Customer environments are unique and these options will allow our customers to look at their space, performance, and capacity needs in new ways.

With the ability to attach so much more DASD to existing towers we want to try to characterize where possible bottlenecks might exist. The first limits are the IOAs and we have attempted to characterize the 571E/574F and 571F/575B in RAID and Mirroring environments. The next limit will be the buses in a single tower. We are using a large file concurrent RSTLIB operations from multiple virtual tape drives located on the DASD in the target HSL loop, to try to help characterize the Bus and HSL limits. The tower is by itself in a single HSL loop, with all the DASD configured into a single user ASP, and RAID5 activated on the IOAs.

As the scenarios progress 2 then 3 towers are added up to 6 in the HSL loop. All 6 have 3 571E/574F's controlling the 45 DASD in the 5094 towers and 3 571F/575B IOAs controlling 108 DASD in #5786 EXP24 Disk Drawer. Multiple Virtual tape drives were created in the user ASP. The 3 other HSL loops contained the system ASP where the data is written to. We used three HSL loops to prevent the destination ASP from being the bottleneck. The system was a 570 ML16 way with 256 GB of memory and originating ASP contained 916 DASD units on 571E/574F and 571F/575B IOAs. Restoring from the virtual tape would create runs of 100% reads from the ASP on the single loop. The charts show the maximum throughput we were able to achieve from this workload.

NOTE: This is a DASD only workload. No other IOAs such as communication IOAs were present.





4.1.6 Direct Attach 571E/574F and 571F/575B Observations

We did some simple comparison measurements to provide graphical examples for customers to observe characteristics of new hardware. We collected performance data using Collection Services and Performance Explorer to create our graphs after running our DASD IO workload (small block reads and writes).

IOP vs IOPLess: no measurable difference in CPU or throughput.

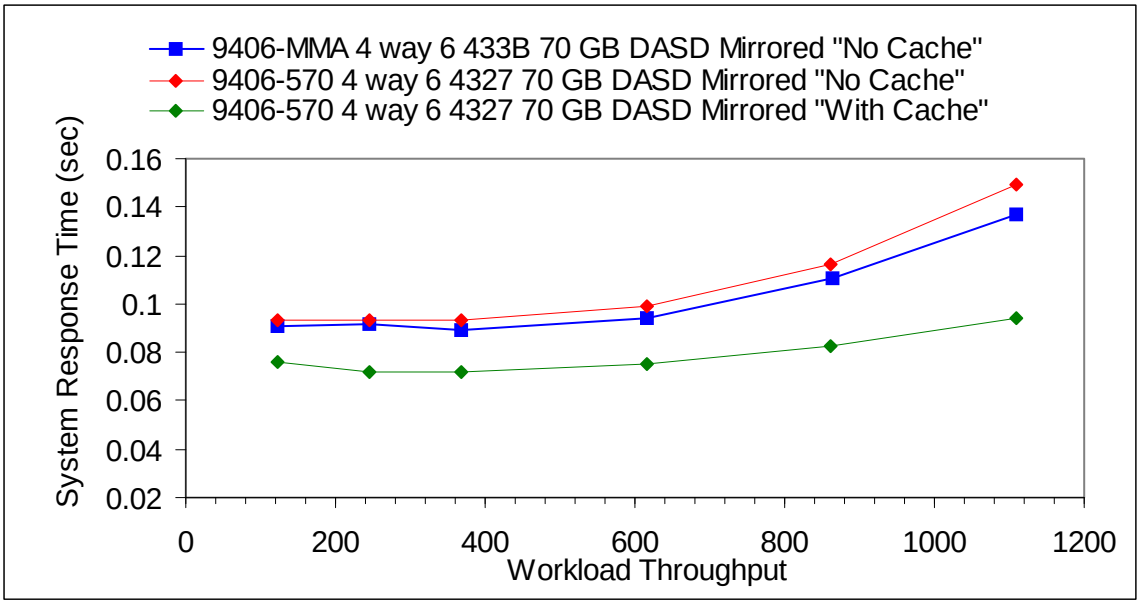
Newer models of DASD are U320 capable and with the new IOAs can improve workload throughput with the same number of DASD or even less in some workload situations.

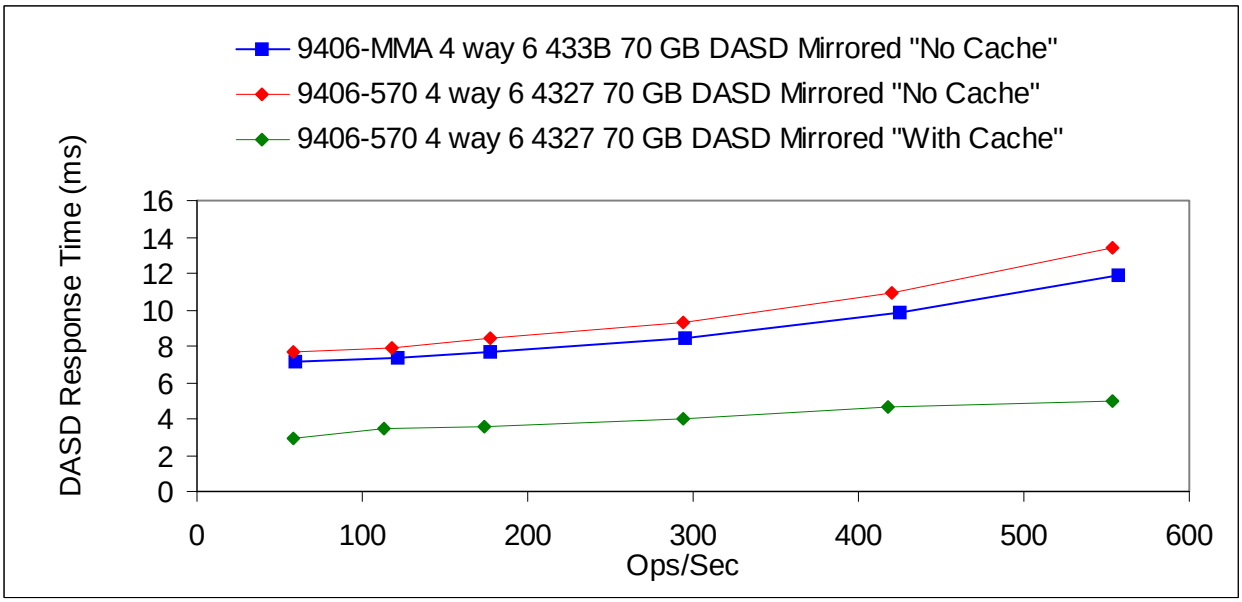
IOA's 571E/574F and the 571F/575B achieved up to 25% better throughput at the 40% DASD Subsystem Utilization point than the 2780/574F IOA. The 571E/574F and 2780/574F were measured with 15 DASD units in a 5094 enclosure. The 571F/575B IOAs attached to #5786 EXP24 Disk Drawers.

System Models and Enclosures: Although an enclosure supports the new DASD or new IOA, you must ensure the system is configured optimally to achieve the increased performance documented above. This is because some card slots or backplanes may only support the PCI protocol versus the PCI-X protocol. Your performance can vary significantly from our documentation depending upon device placement. For more information on card placement rules see the following link:
IBM i operating system iV5R2: http://www.redbooks.ibm.com/redpapers/pdfs/redp3638.pdf
IBM i operating system iV5R3, iV5R4:
http://www.redbooks.ibm.com/redpapers/pdfs/redp4011.pdf

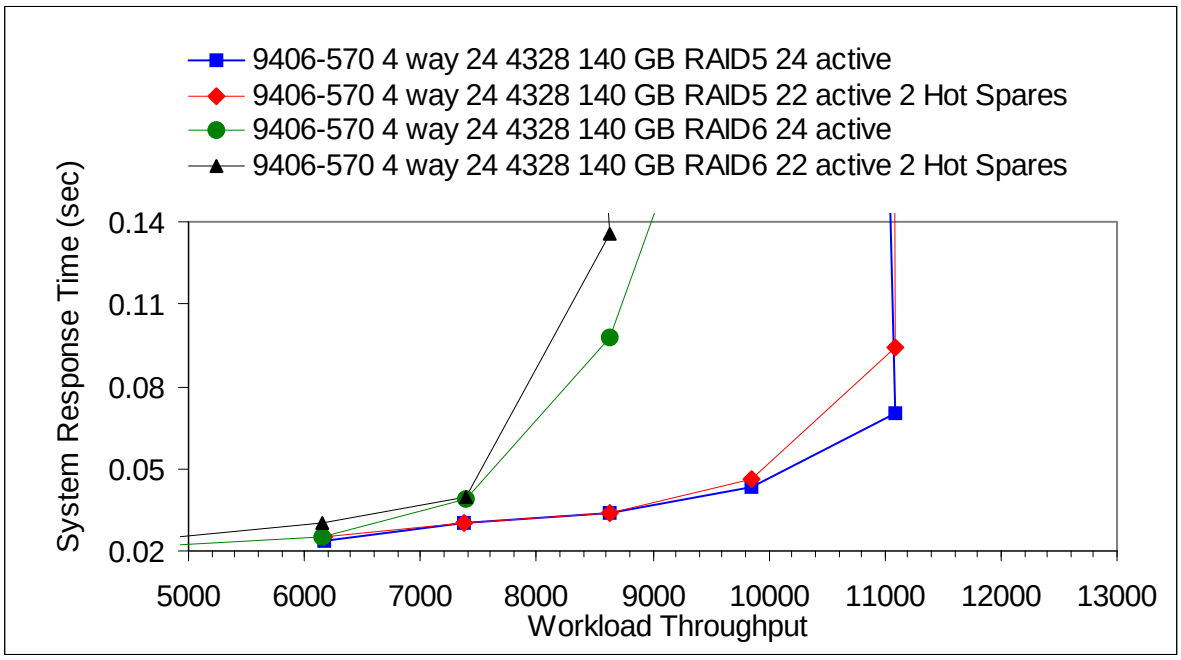
Conclusions: There can be great benefits in updating to new hardware depending upon the system workload. Most DASD intense workloads should benefit from the new IOAs available. Large block operations will greatly benefit from the 5094/5294 feature code #6417/9517 enclosures in combination with the new IOA's and DASD units.
Note: The #6417/9517 provides a faster HSL-2 interface compared to the #2887/9877 and is available for I/O attached to POWER-5 based systems

4.1.7 9406-MMA CEC vs 9406-570 CEC DASD

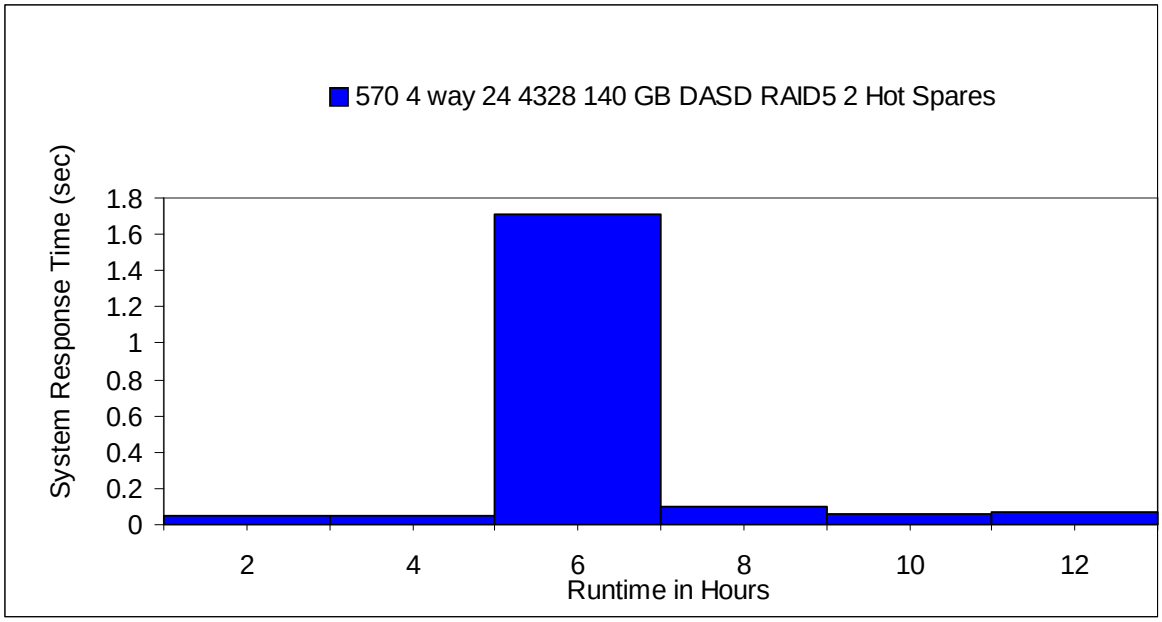




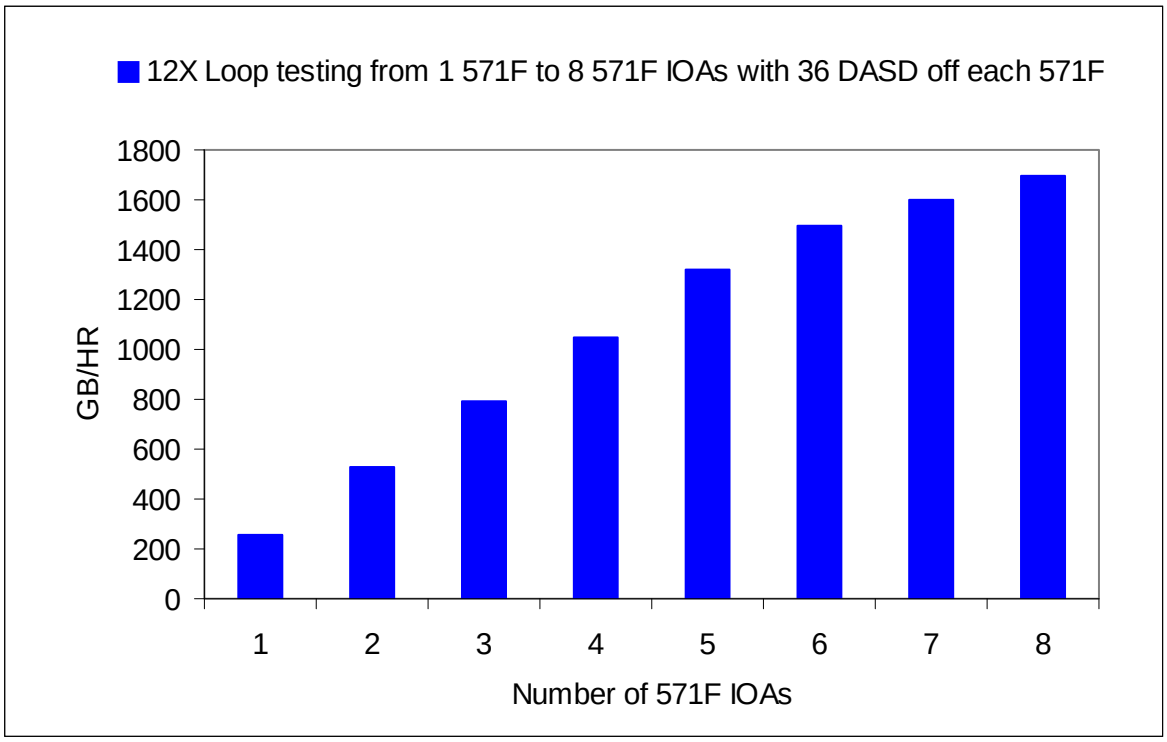
4.1.8 RAID Hot Spare



For the following test, the IO workload was setup to run for 14 hours. About 5 hours after starting A DASD was pulled from the configurations. This forced a RAID set rebuild.



4.1.9 12X Loop Testing

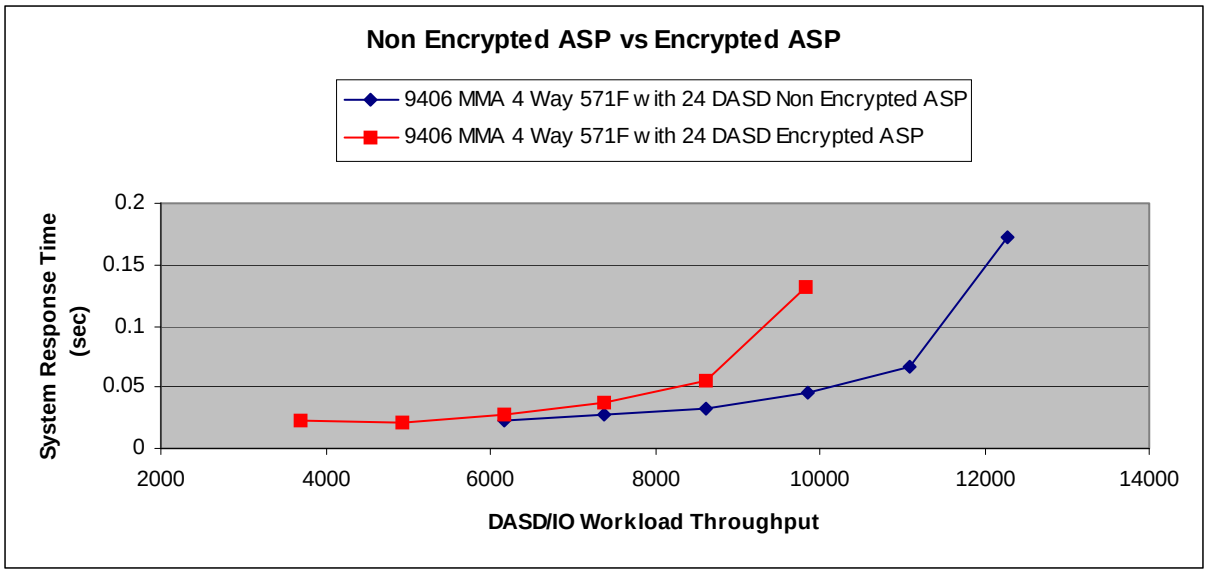


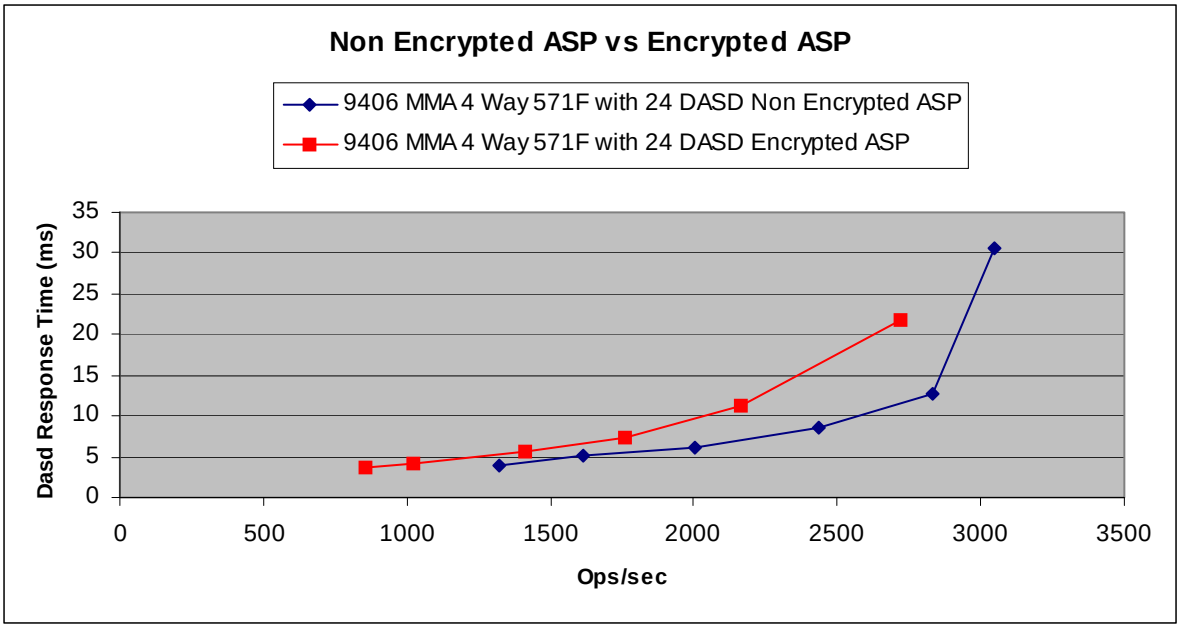
A 9406-MMA 8 Way system with 96 GB of mainstore and 396 DASD in #5786 EXP24 Disk Drawer on 3 12X loops for the system ASP were used, ASP 2 was created on a 4th 12X loop by adding 5796 system expansion units with 571F IOAs attaching 36 4327 70 GB DASD in #5786 EXP24 Disk Drawer with RAID5 turned on. we created a virtual tape drive in ASP2 and we used a 320GB file to save to the tape drive for this test.

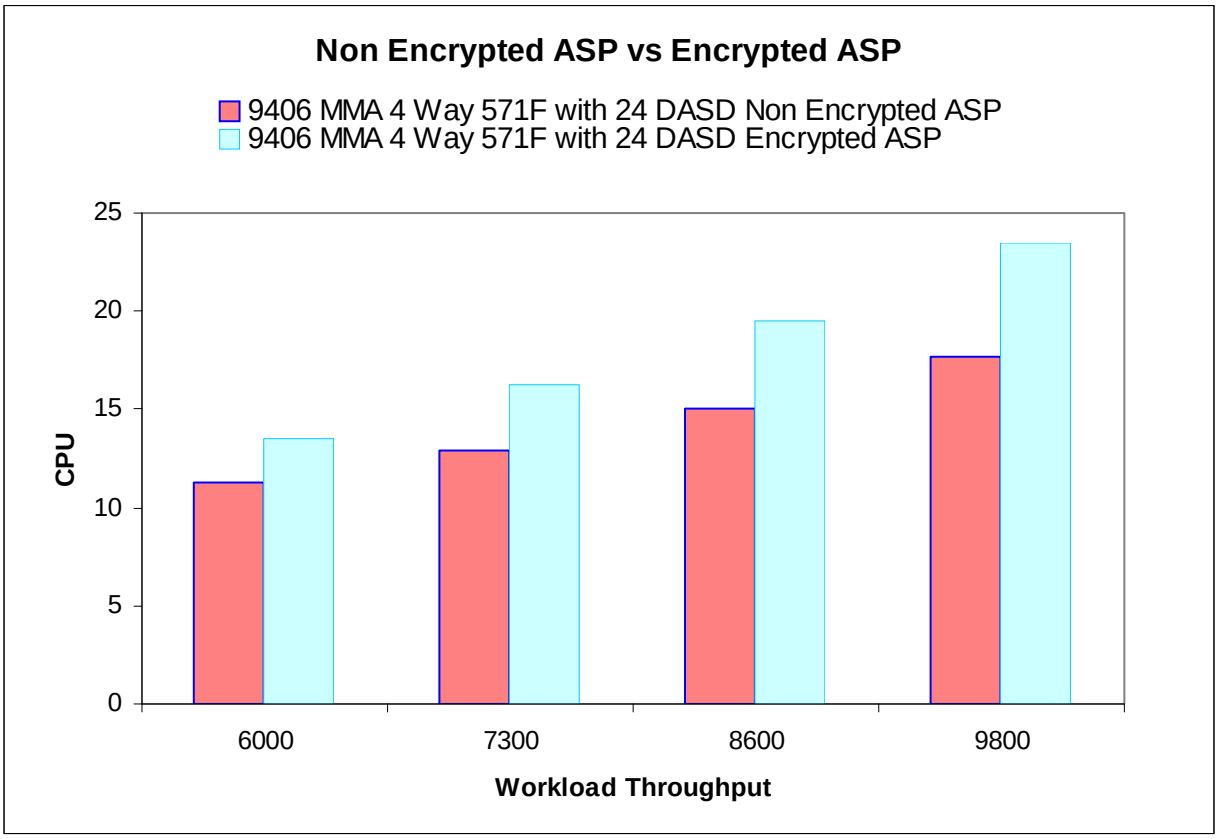
When we completed the testing up to 288 DASD on 8 IOAs, we moved the 12X loop to the other 12X GX adapter in the CEC and ran the test again and saw no difference in the testing between the two loops. The 12X loop is rated for more throughput than the DASD configuration would allow for. So the test isn't a tell all about the 12X loops capabilities only a statement of support to the maximum number of 571F IOAs allowed in the loop.

4.1.10 V6R1M0 Encrypted ASP

More CPU and memory may be needed to achieve the same performance once encryption is enabled.

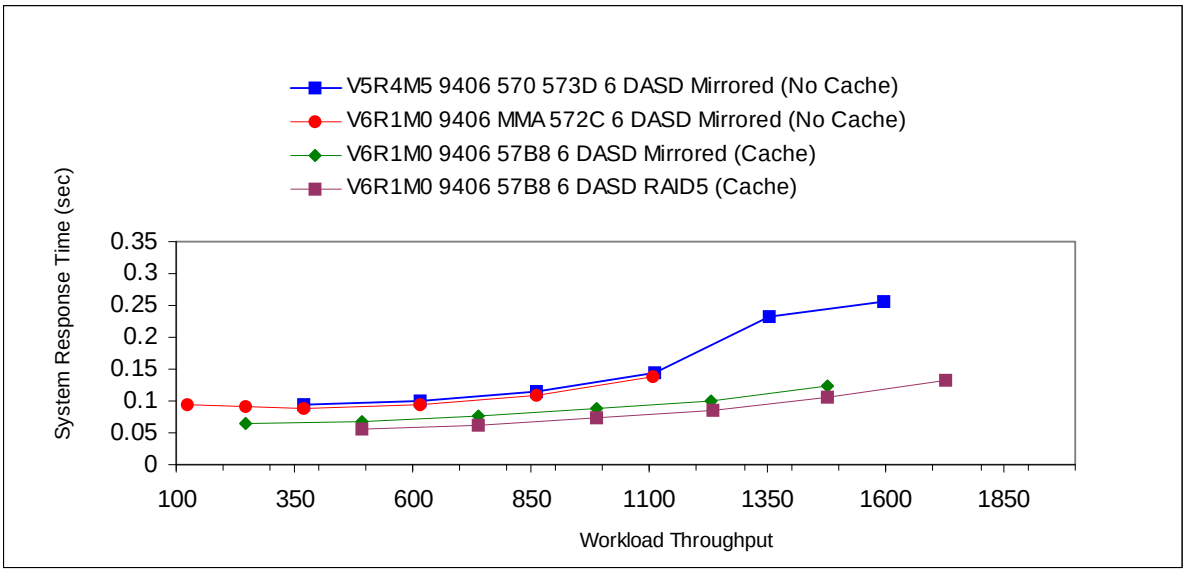


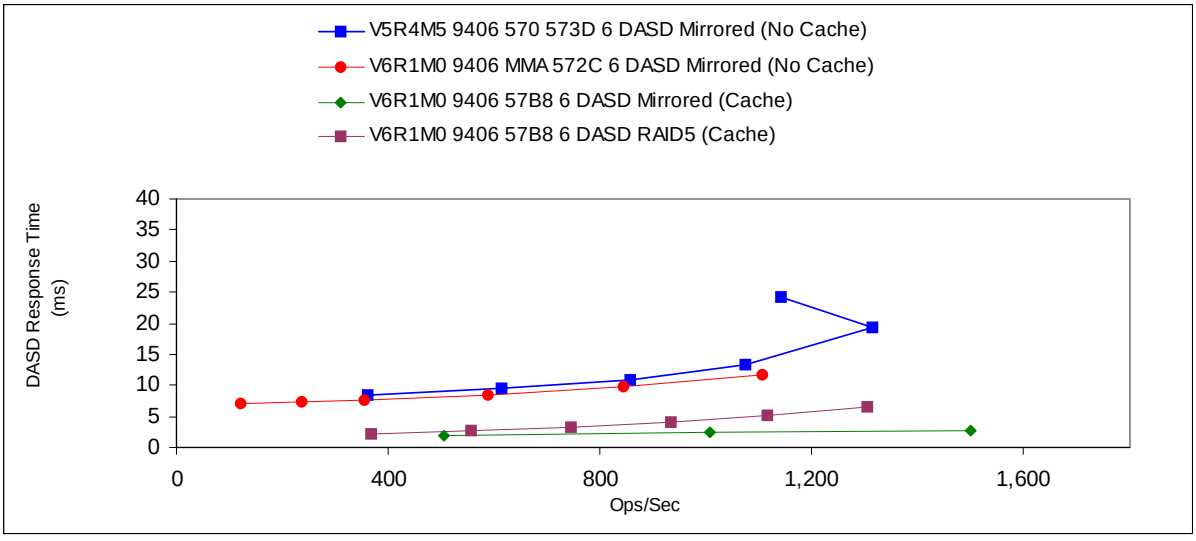




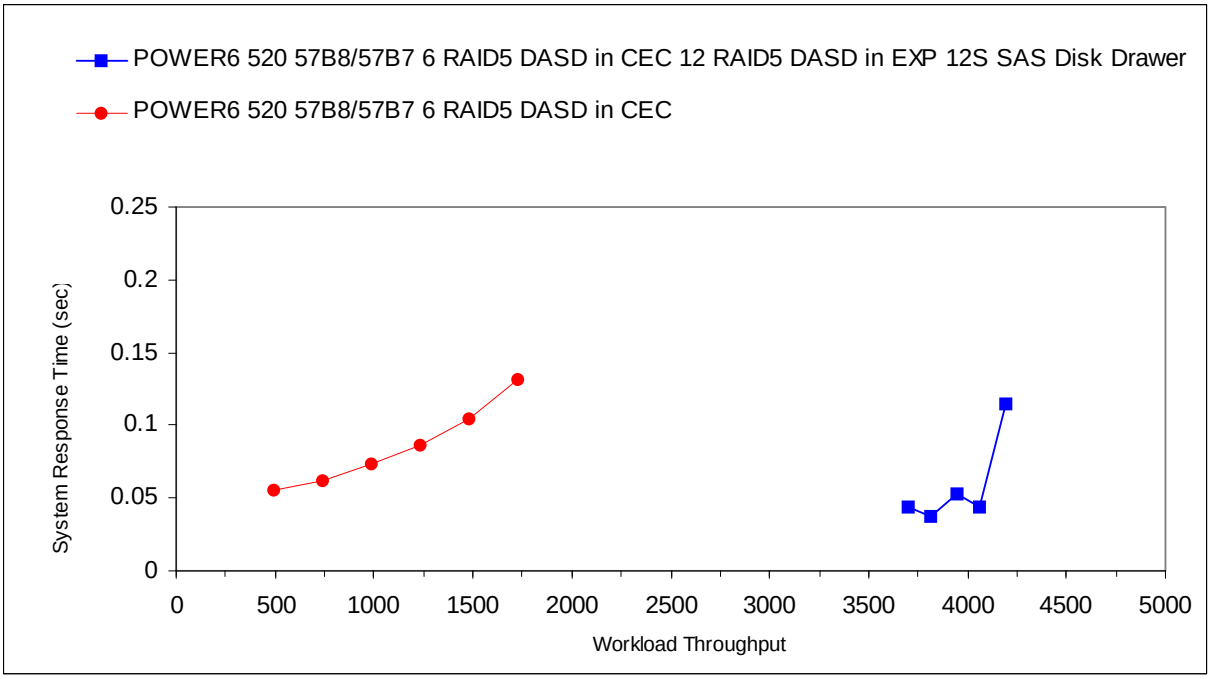
4.1.11 57B8/57B7 IOA

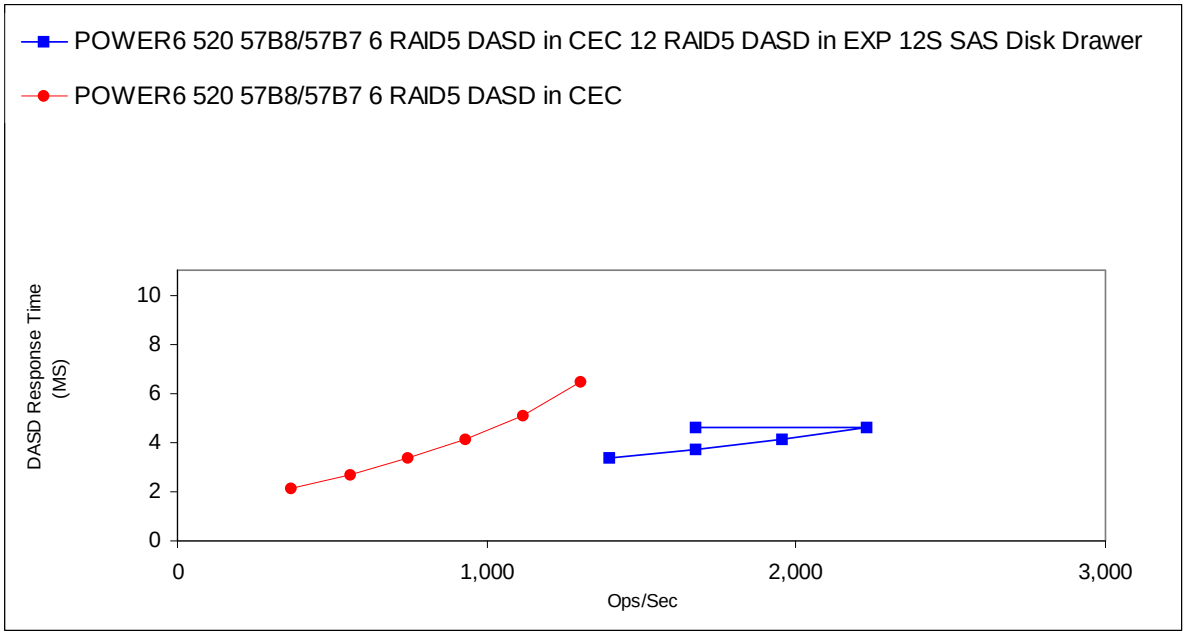
With the addition of the POWER6 520 and 550 systems comes the new 57B8/57B7 SAS Raid enablement controller with Auxiliary Write Cache. This controller is only available in the POWER6 520 and 550 systems and provides RAID5/6 capabilities, with 175MB redundant write cache. Below are some charts comparing the Storage Controllers for the POWER5 570 (573D), which can be either mirrored or RAID5 protected. The POWER6 570 (572C) which can only be mirrored, and the POWER6 520/550 (57B8/57B7) which can be RAID5/6 or protected with mirroring.





The POWER6 520 and 550 also have an external SAS port, that is controlled by the 57B8/57B7, used to connect a single #5886 - EXP 12S SAS Disk Drawer which can contain up to 12 SAS DASD. Below is a chart showing the addition of the #5886 - EXP 12S SAS Disk Drawer.



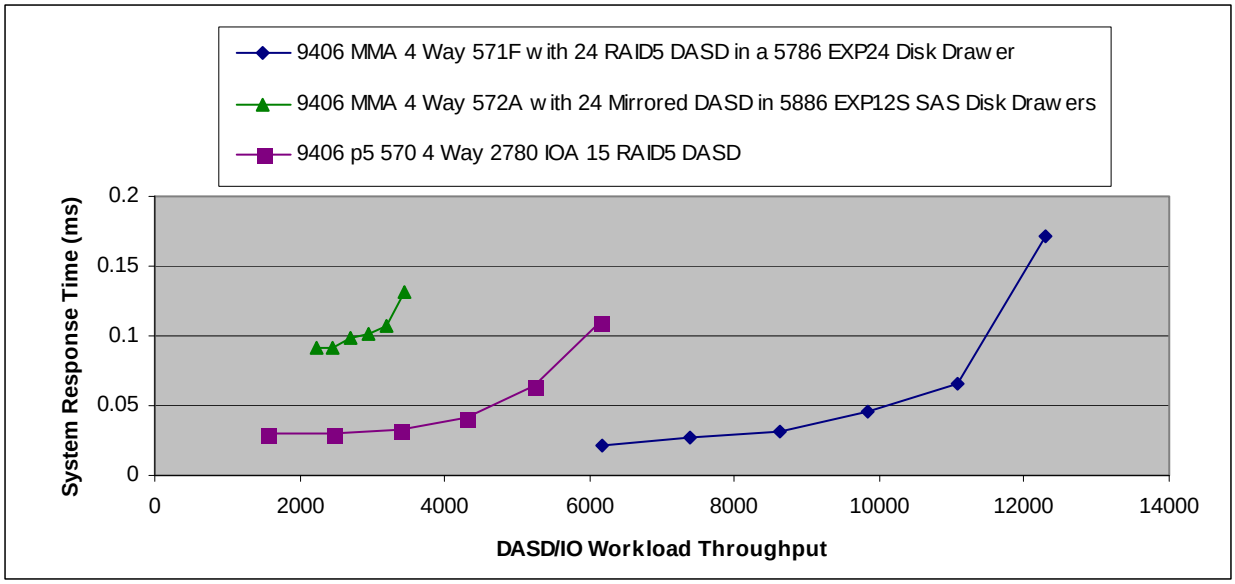


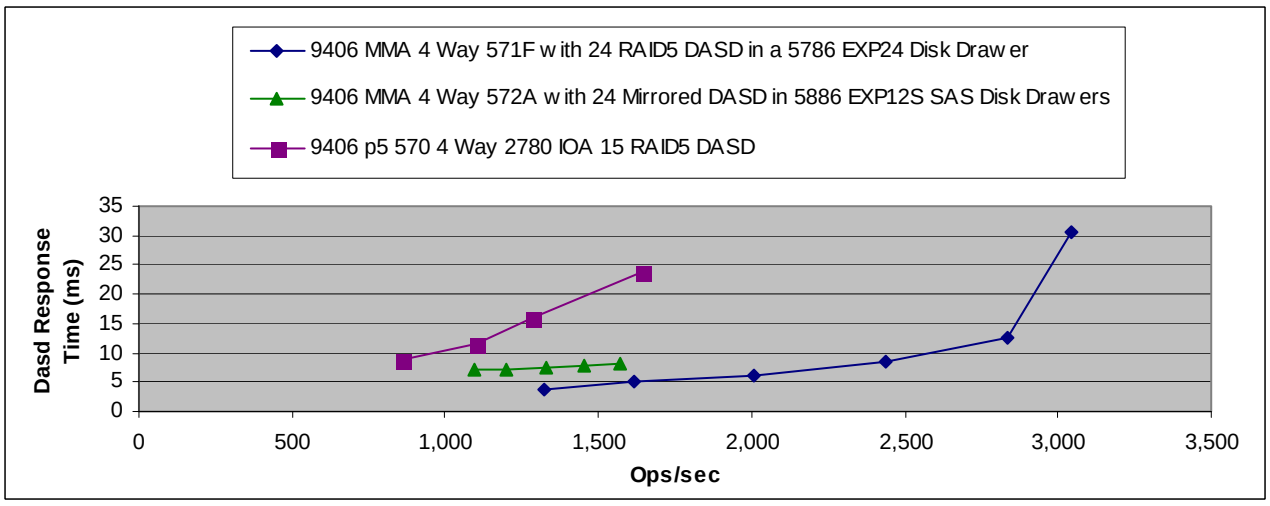
4.1.12 572A IOA

The 572A IOA is a SAS IOA that is mainly used for SAS tape attachment but the 5886 EXP 12S SAS Disk Drawer can also be attached.

This IOA does not have cache, so its performance will be much less than those that do have cache. The following charts help to show the performance characteristics that resulted during experiments in the Rochester lab.

If storage space is all that is needed then the 5886 EXP 12S SAS Disk Drawer could be an option

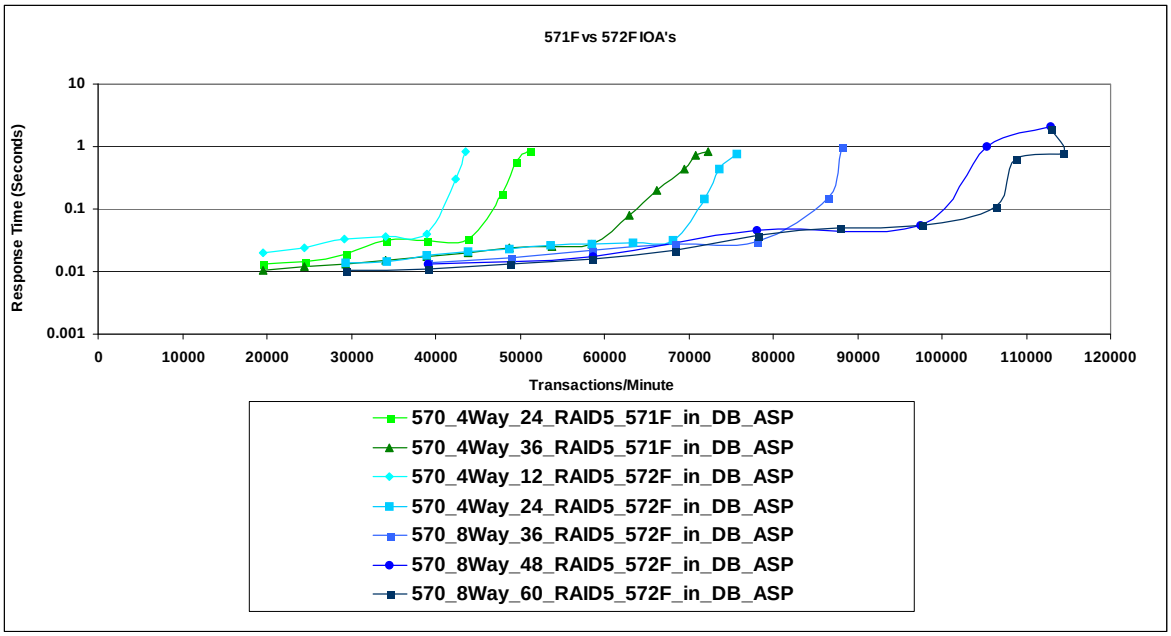


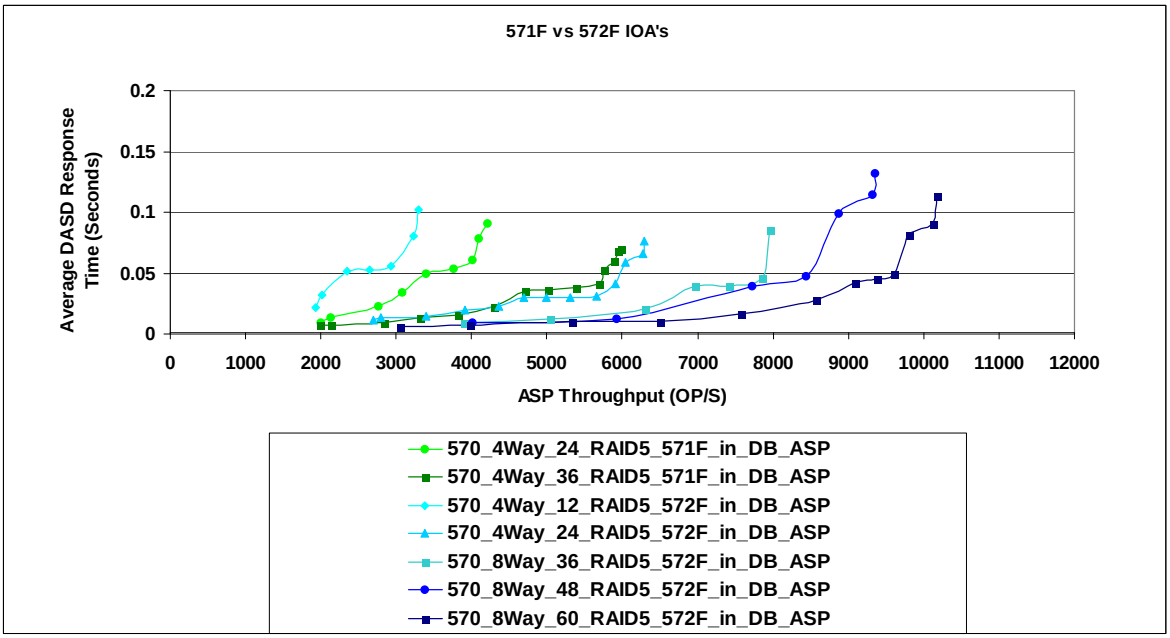


4.1.13 572F IOA with 15K SAS DASD

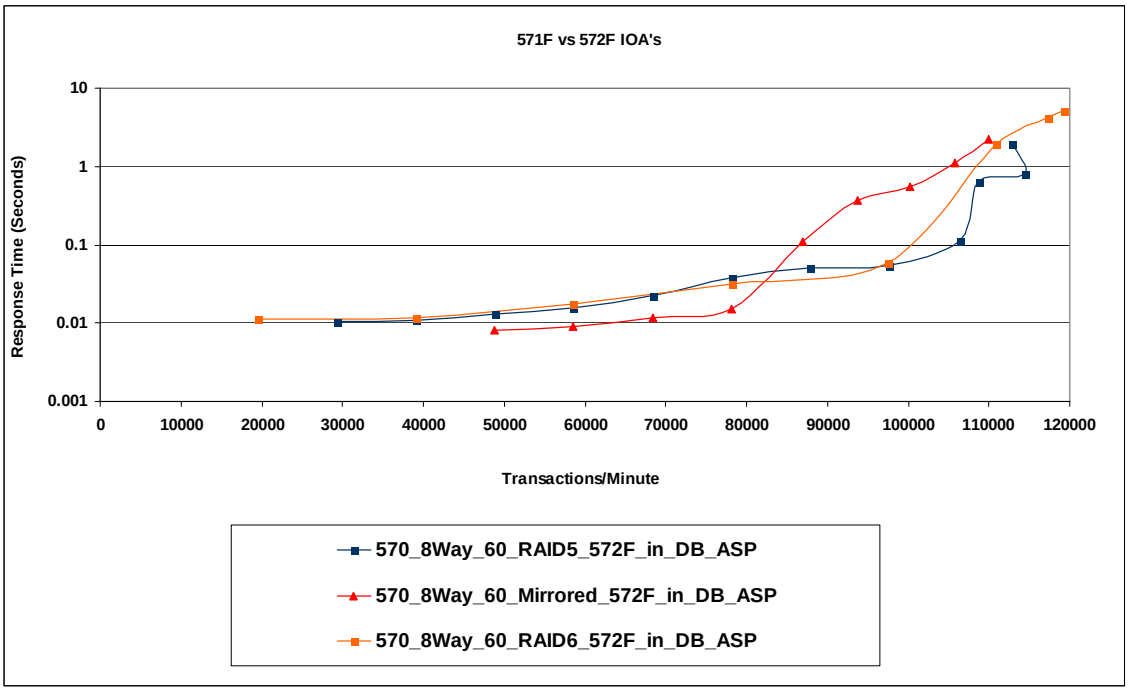
The 572F/575C IOA is being introduced to attach the EXP12 drawers. The following section shows a comparison of a single 571F with 15K RPM SCSI DASD and the 572F with 15K RPM SAS DASD.

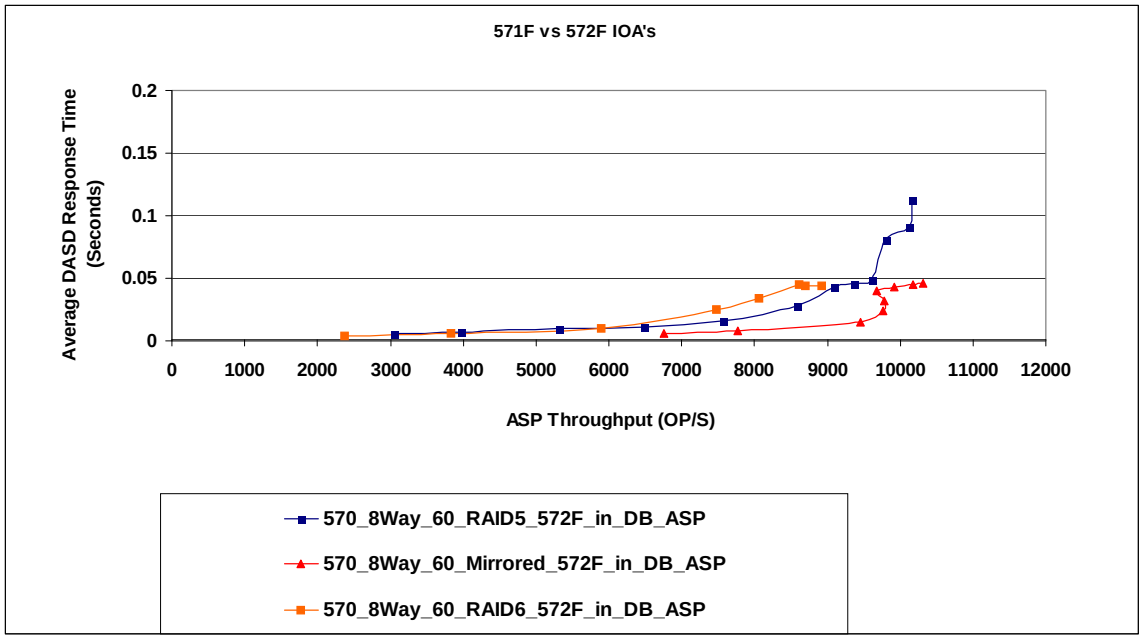
NOTE: A variation of the Commercial Performance Workload was used to create the following charts, the characteristics of this workload are similar to the DASDIO workload but cannot be directly compared.





The following charts compare mirroring, RAID5 and RAID6 on a single 572F/575C IOA. Mirroring provides slightly better performance up to 36 DASD, above 36 to 60 DASD RAID5 and RAID6 appear to make better use of the IOA's capabilities. These environments were set up to show the performance characteristics and are not necessarily recommended for customer environments as the mirrored environment in this example creates a single point of failure.

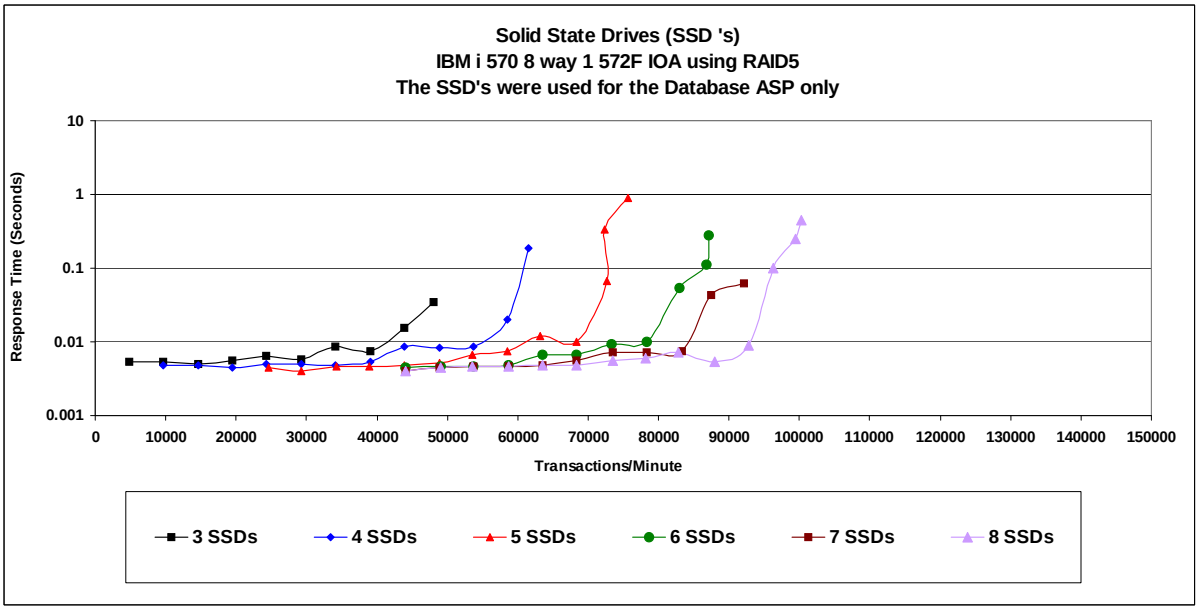


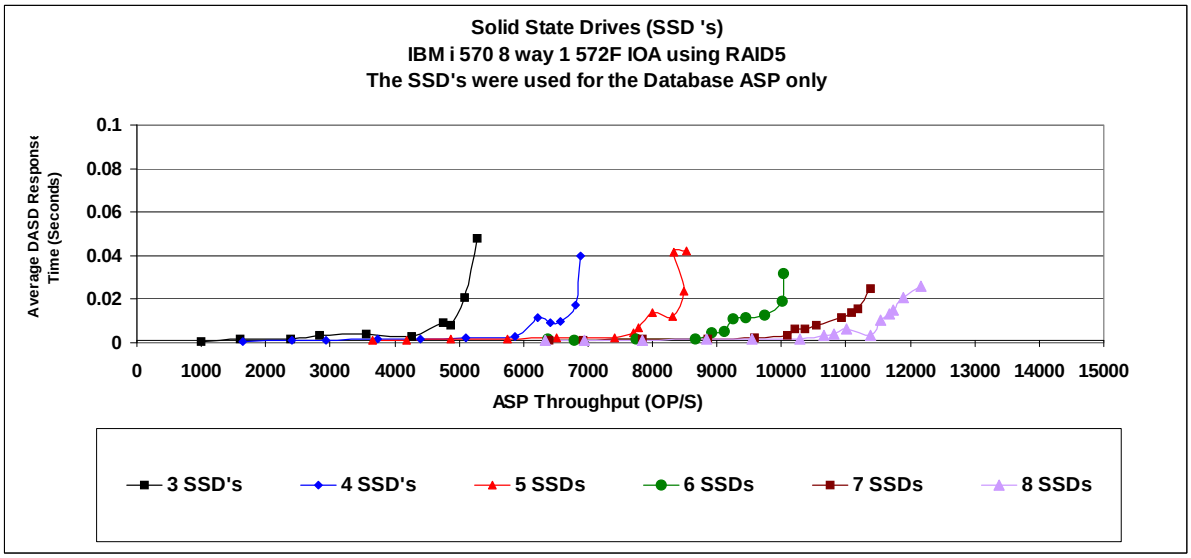


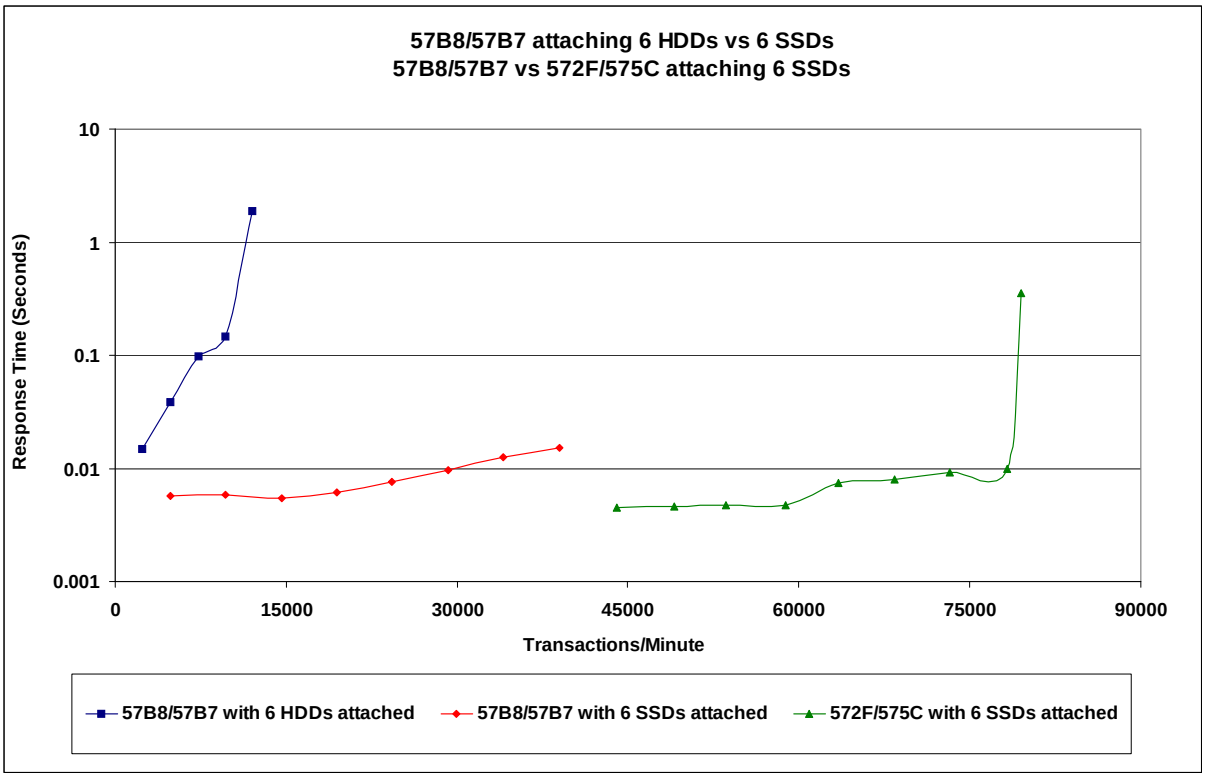
4.1.14 Solid State Drives (SSDs)

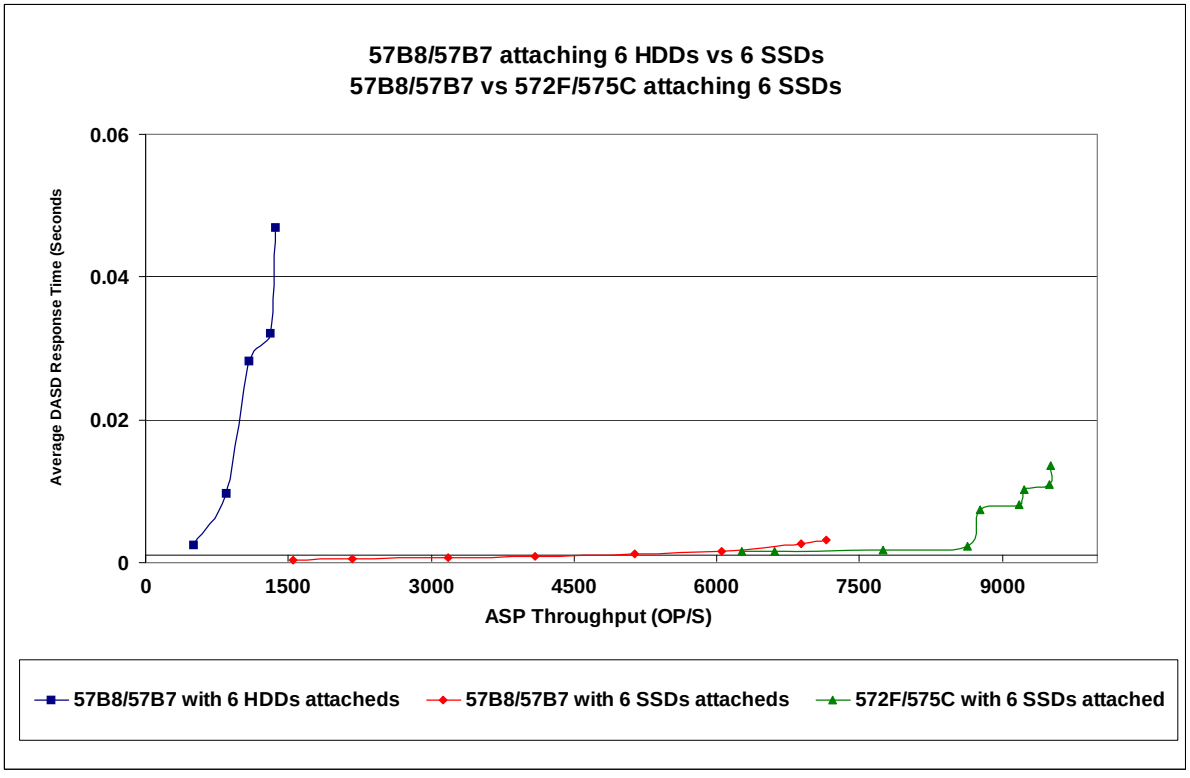
The following charts are experiments the Rochester lab conducted to show the scaling of SSDs on a single 572F/575C IOA. SSDs can be a substantial performance benefit for some applications. To understand the benefits and how to take advantage of SSDs please see the following whitepaper: http://www.ibm.com/systems/resources/ssd_ibmi.pdf

NOTE: A variation of the Commercial Performance Workload was used to create the following charts, the characteristics of this workload are similar to the DASDIO workload but cannot be directly compared.



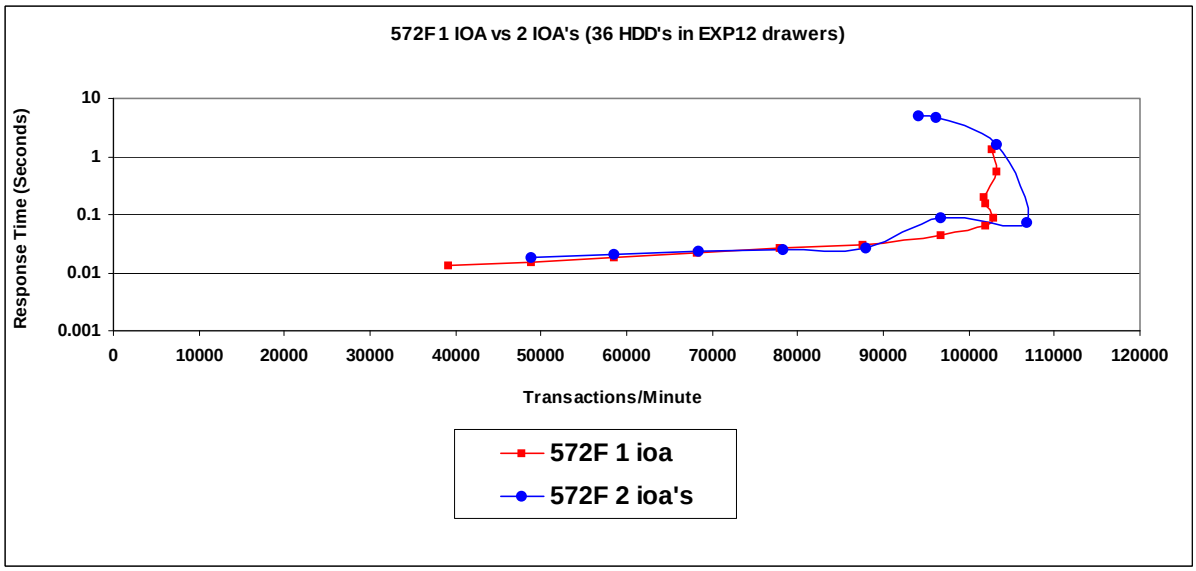


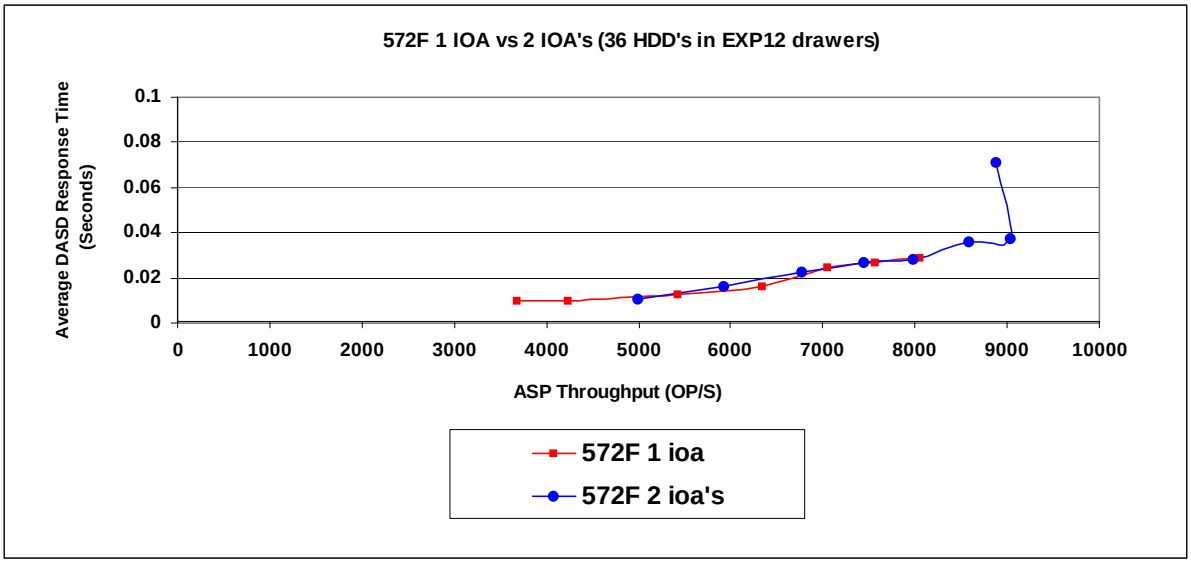




4.1.15 V6R1M1 (574E, SSD SkipOps, Dual IOAs)

The following sections have charts created from data collections on new hardware and the V6R1M1 IBM i OS.
NOTE: In order to attempt to find the limitations on the SSDs in these environments the DASDIO workload was modified so not all charts can be compared directly to others. Notes are included in the sections where the modified workload was used.

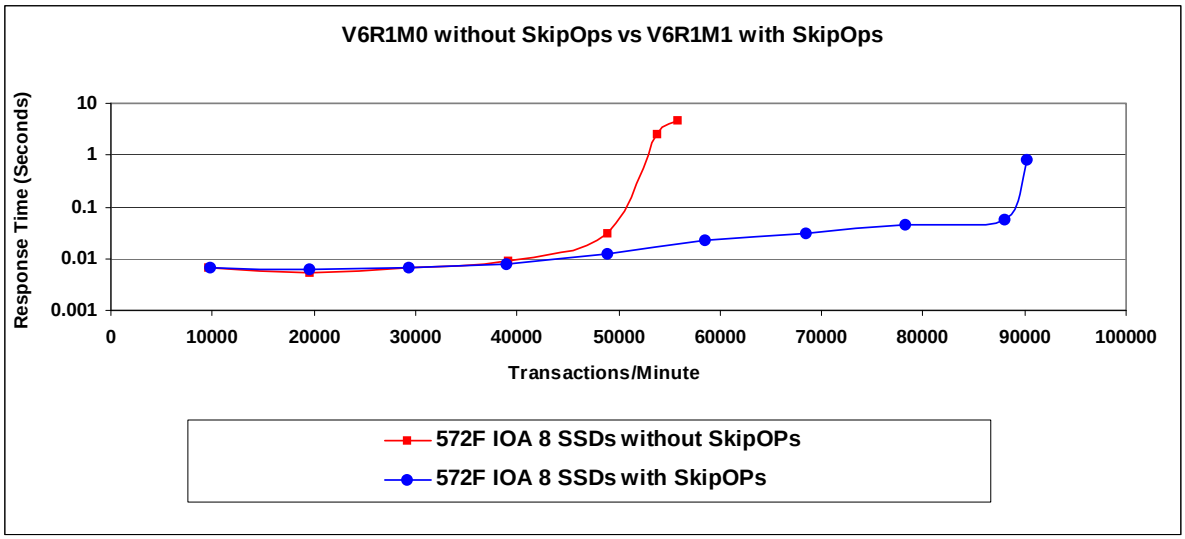


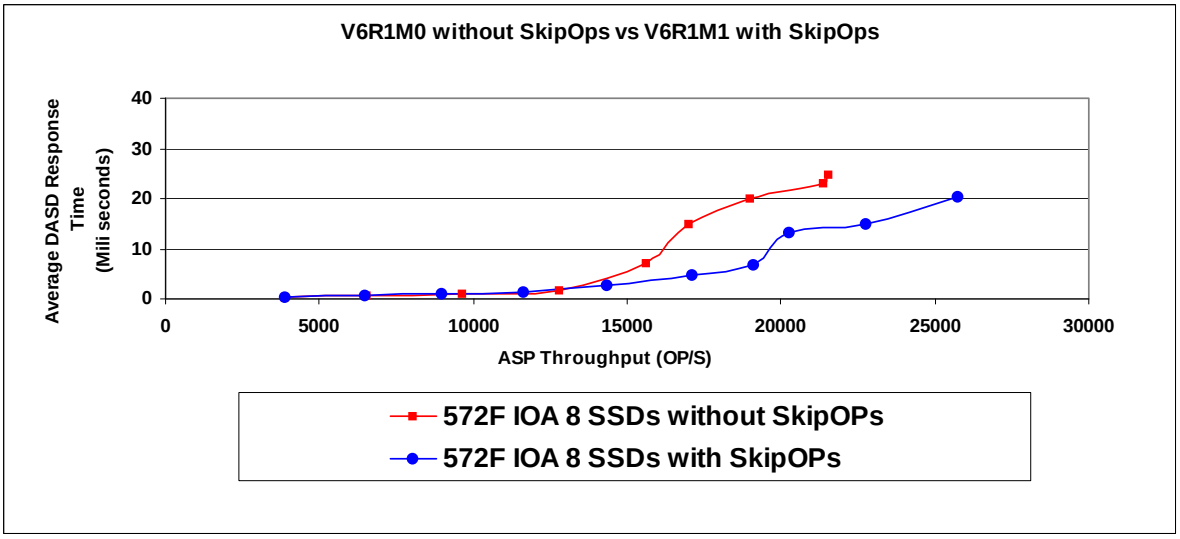


The previous 2 charts show the 572F in a single IOA configuration and a dual IOA configuration, with HDDs. Both environments performed the same so this configuration doesn't enhance performance. But for those wanting a stronger protection enviroment , the dual IOA maybe a good fit .

The following charts show the performance boost available when moving to V6R1M1 with the 572F IOA and SSD's.

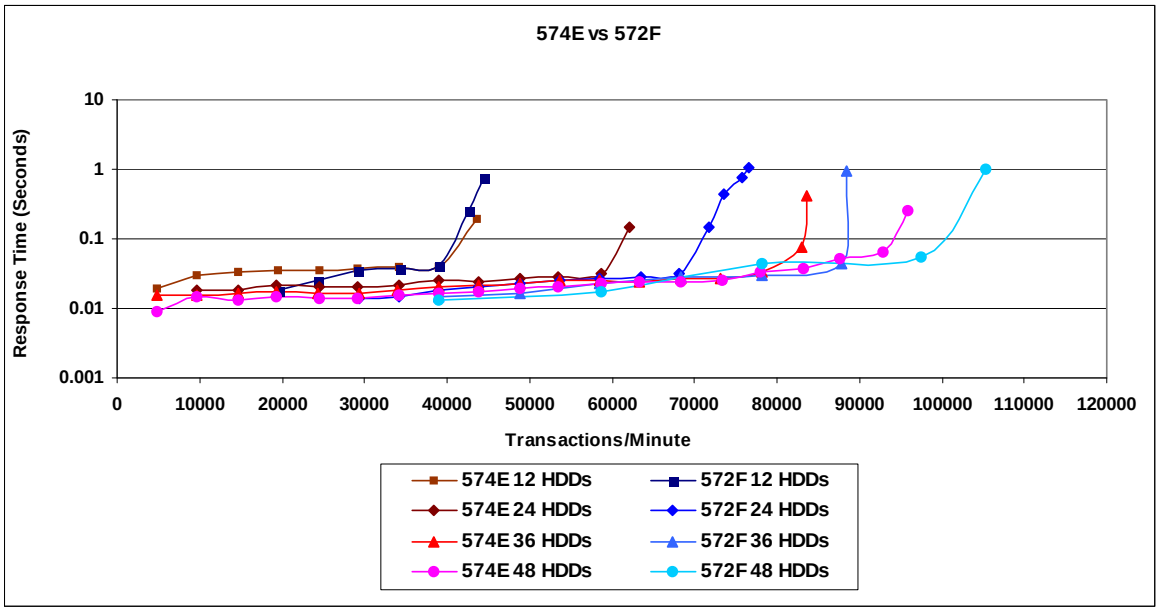
For SSD runs it was necessary to change the DASDIO workload in order for us to get the workload to run enough operations through the IOAs and SSDs to see where the bends are. So the following charts can't be directly compared with previous DASDIO workload charts. The system memory was decreased by 2/3s in order to force more paging and write activity to the devices.

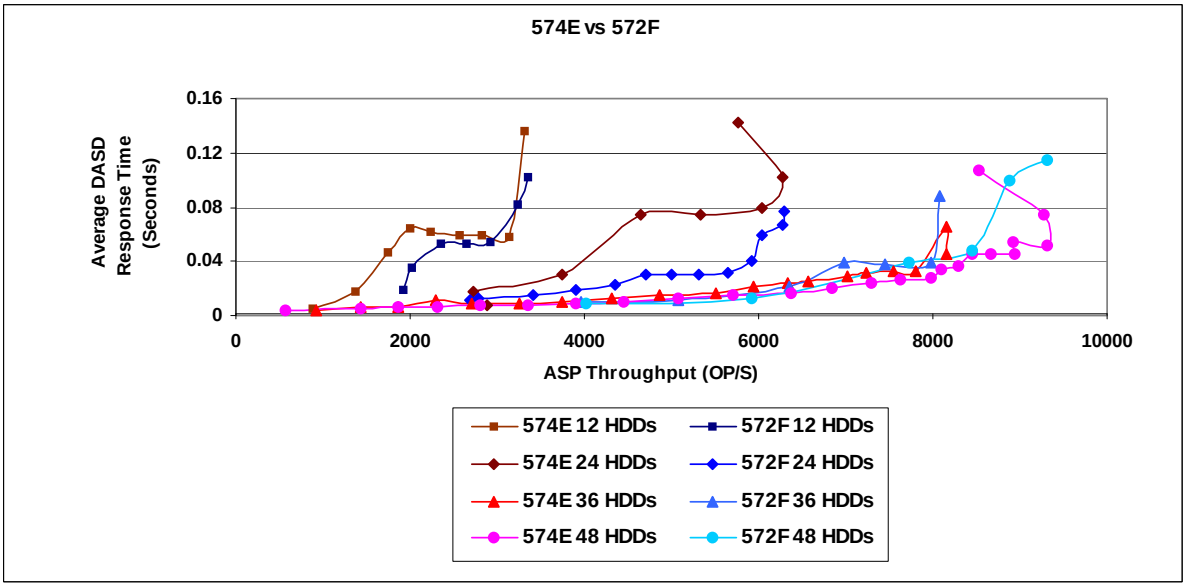




4.1.16 574E IOA

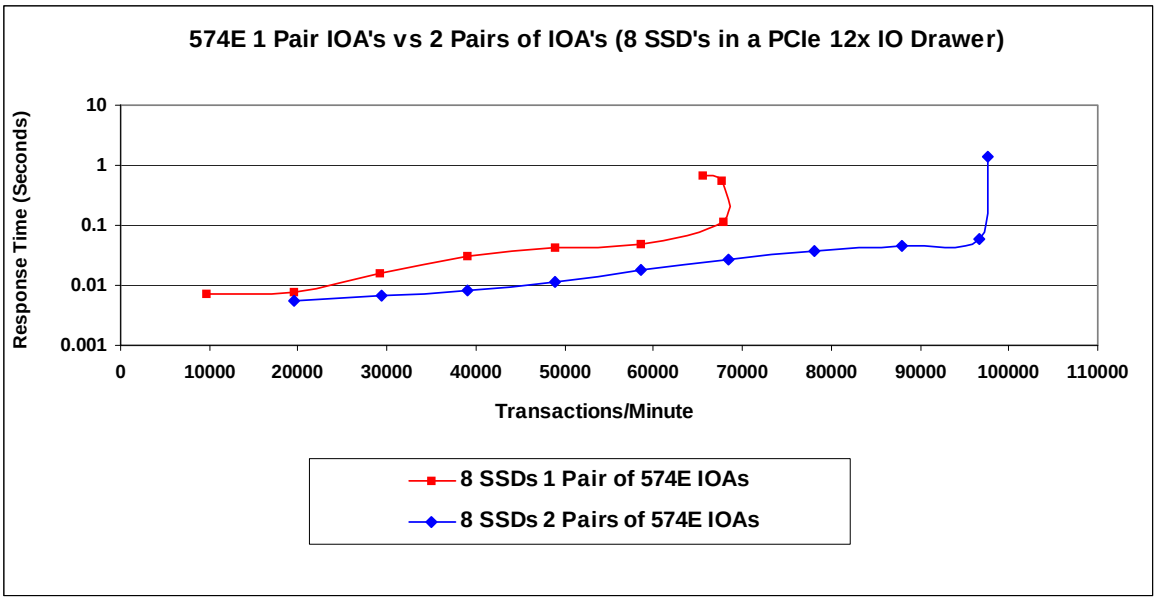
This section shows workload comparisons between the 574E IOA and the 572F. The 574E must always be used in pairs. When there are multiple RAID arrays on the IOA it takes advantage of the **Active/Active** dual path software enhancements in V6R1M1. This is a performance comparison and each customer must weigh the value of cache size and the number of devices that each IOA supports to make the decision as to which one is right in their environment.

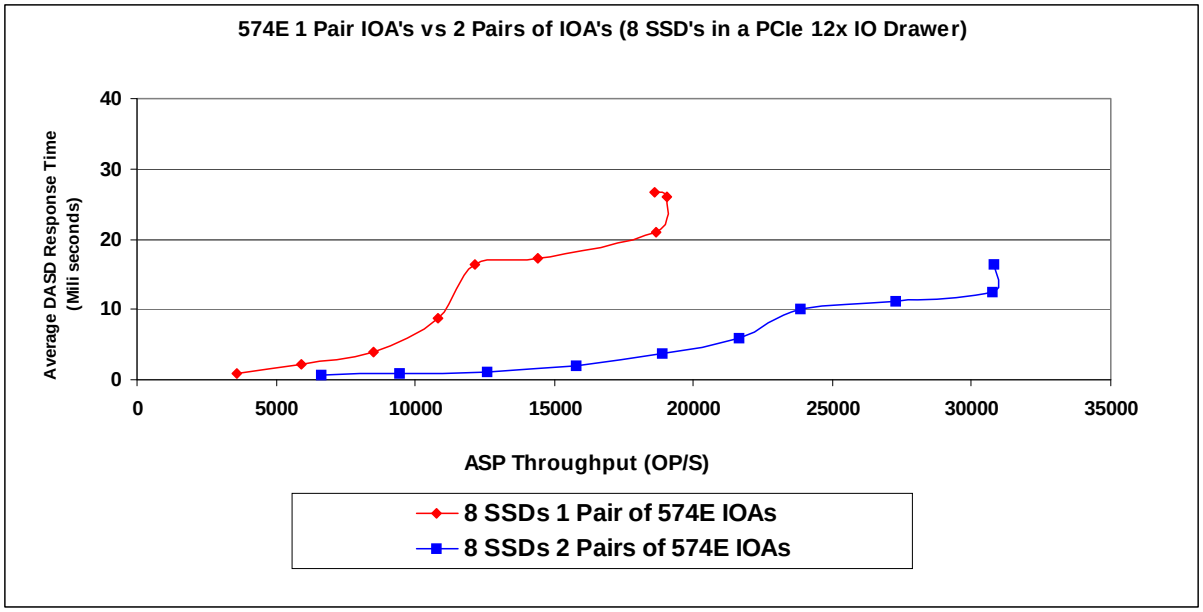




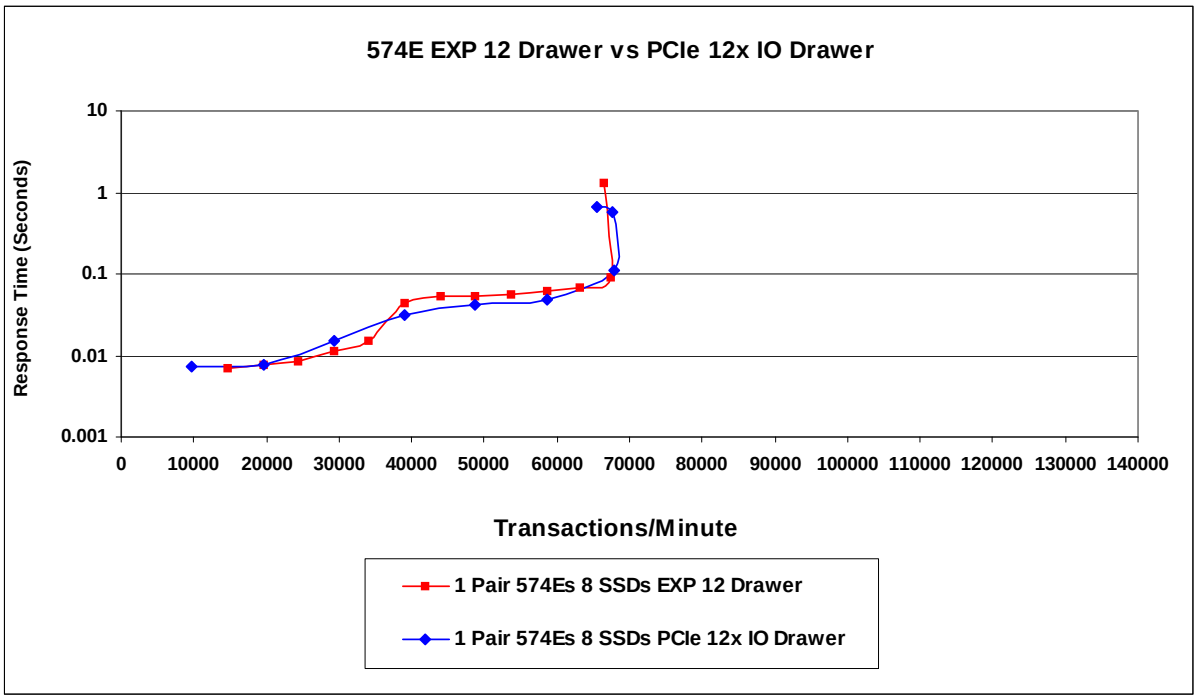
4.1.17 574E 1 pair of IOAs vs 2 pair of IOAs with 8 SSDs

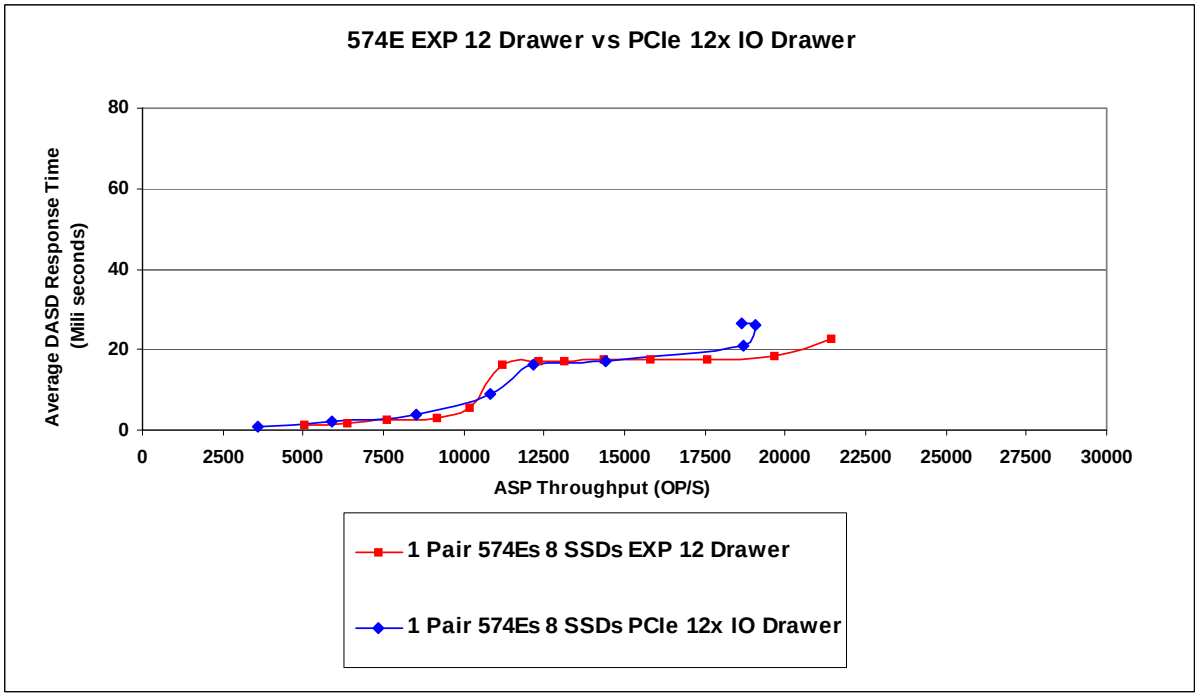
The PCIe 12x IO drawer holds 18 devices and can be split onto 2 IOAs. Even though the drawer and a single pair of IOAs can support up to 9 SSDs in this environment, to get the most performance out of the devices, one might want to consider 4 devices per IOA pair.
NOTE: the SSD data collections used the modified DASDIO workload.





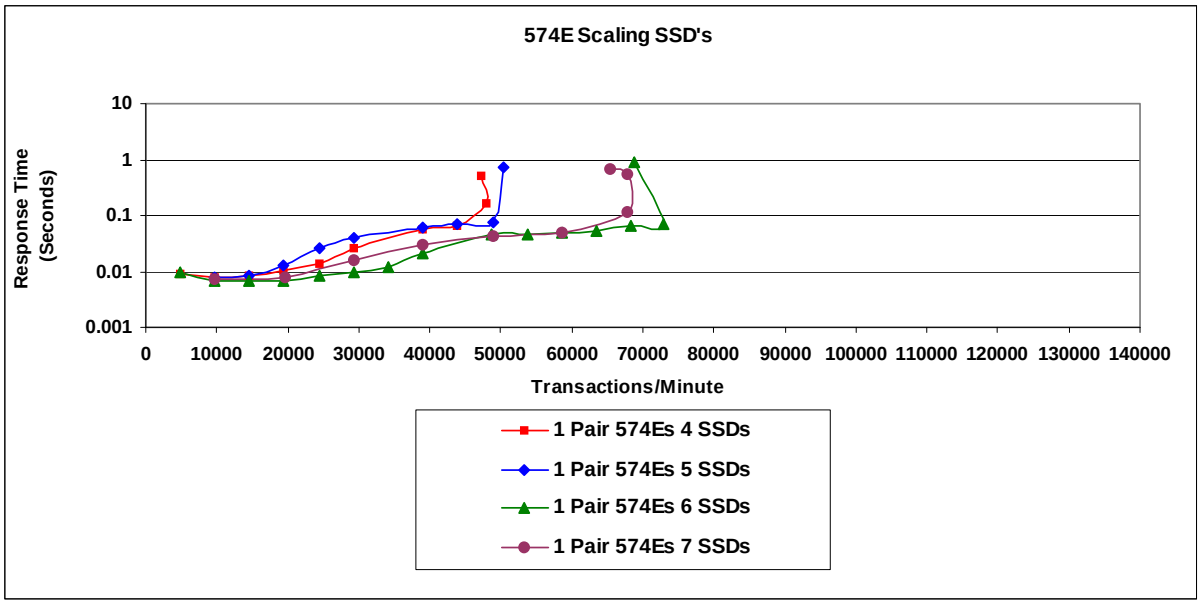
4.1.18 EXP 12 Drawer vs PCIe 12x IO Drawer

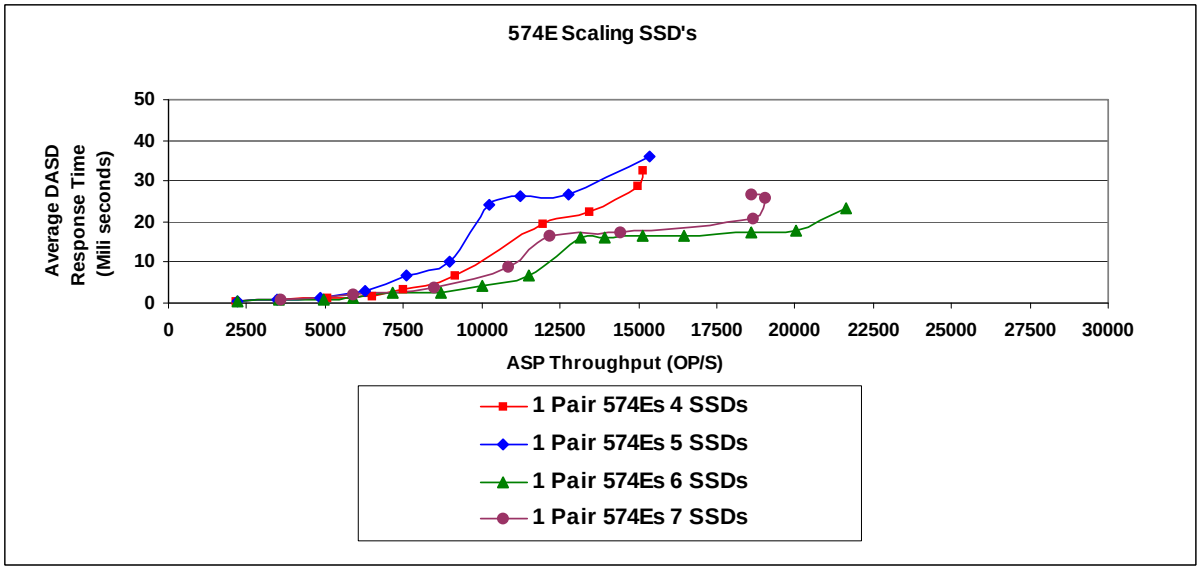




4.1.19 574E SSD scaling

The interesting item in the following charts is the jump from 5 SSDs to 6 SSDs. When the RAID optimization is set for performance with up to 5 SSDs 1 RAID set is created. This means one path from the 574E is the Active path and one is a passive path or failover path if you will. When you have 6 or more devices you get multiple RAID sets. The 'Primary' 574E paths will be the Active path for one RAID set and the 'Secondary' 574E will be the Active path for the other RAID set. Conversely, each of the two 574E's will be the Passive path for the others Active path RAID set.





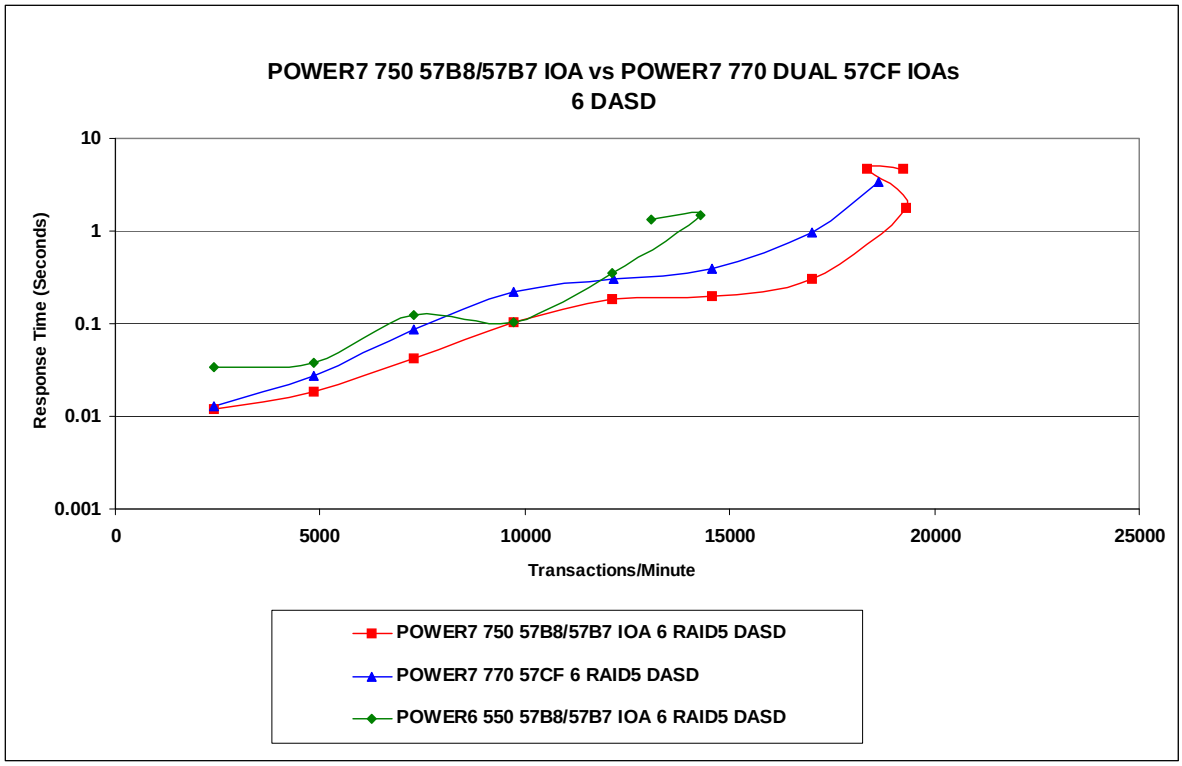
4.1.20 POWER7 750 57B8/57B7 IOA and POWER7 770 57CF IOA and storage expansion port.

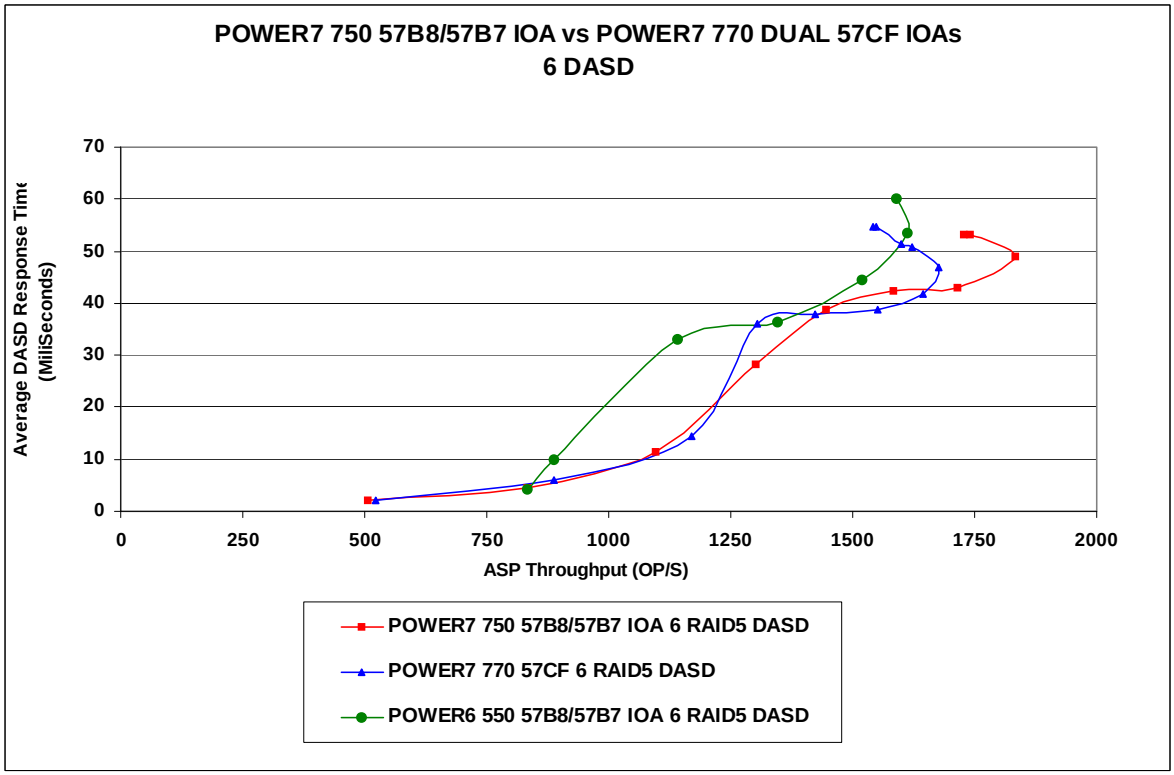
The following charts are results seen in experiments with the 57B8/57B7 and 57CF IOA attaching 6 DASD and 18 DASD.

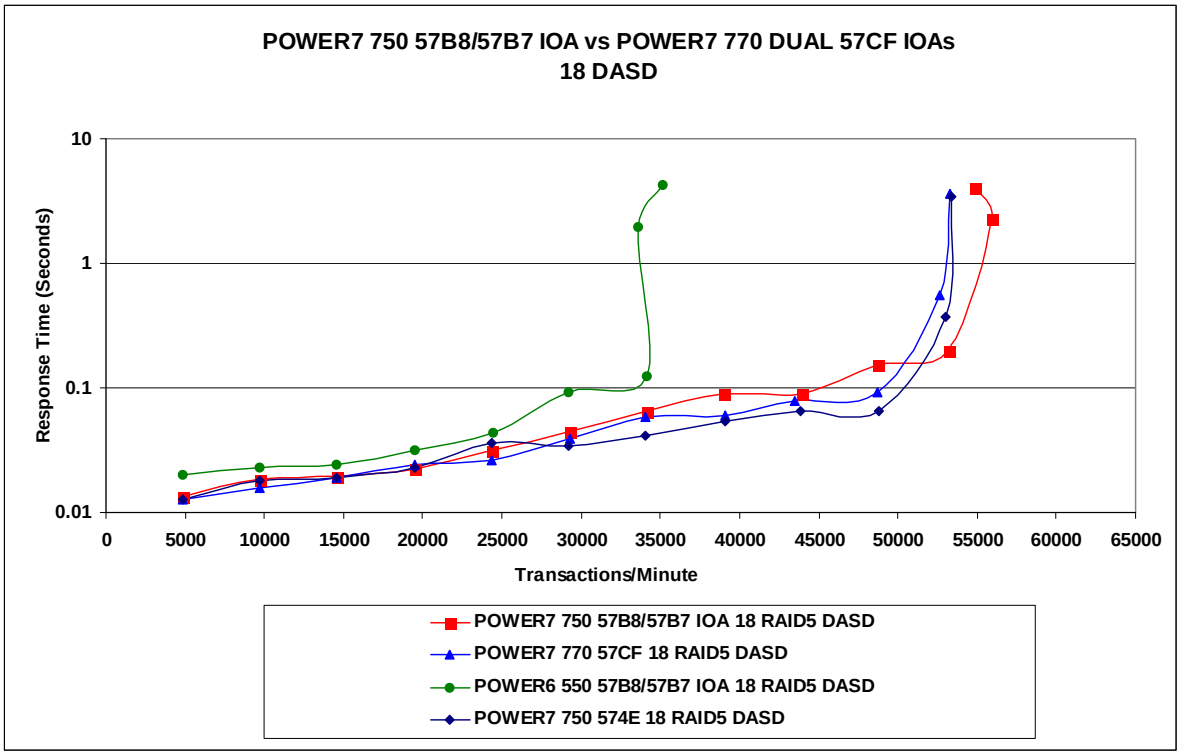
The 57B8/57B7 would actually support 8 and 20 DASD but for comparisons we chose to only install 6 SFF devices in the CEC for these experiments so the results could be compared.

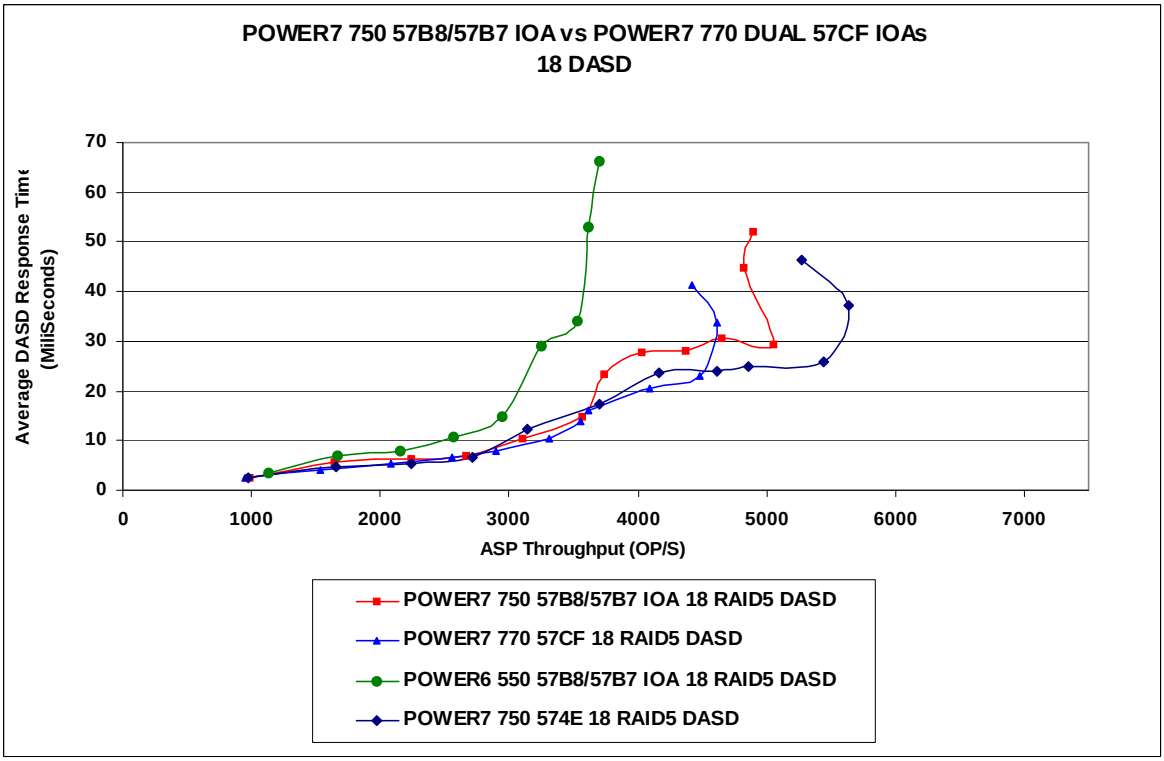
Normally a separate journal ASP is created so that we look only at the database results but because of the constrained resources present with so few DASD the System, Database and the Journal all share the same ASP.

This results in a higher write ratio than what is depicted in most of our other DASDIO workload charts. The workload is usually 60% read and 40% write but because of the journal activity in this experiment that is reversed resulting in a 40% read and 60% write workload. The higher write ratio fairs slightly better in the single IOA environment because there is more simplistic data mirroring structure and no data balancing activity like the DUAL IOA environment.

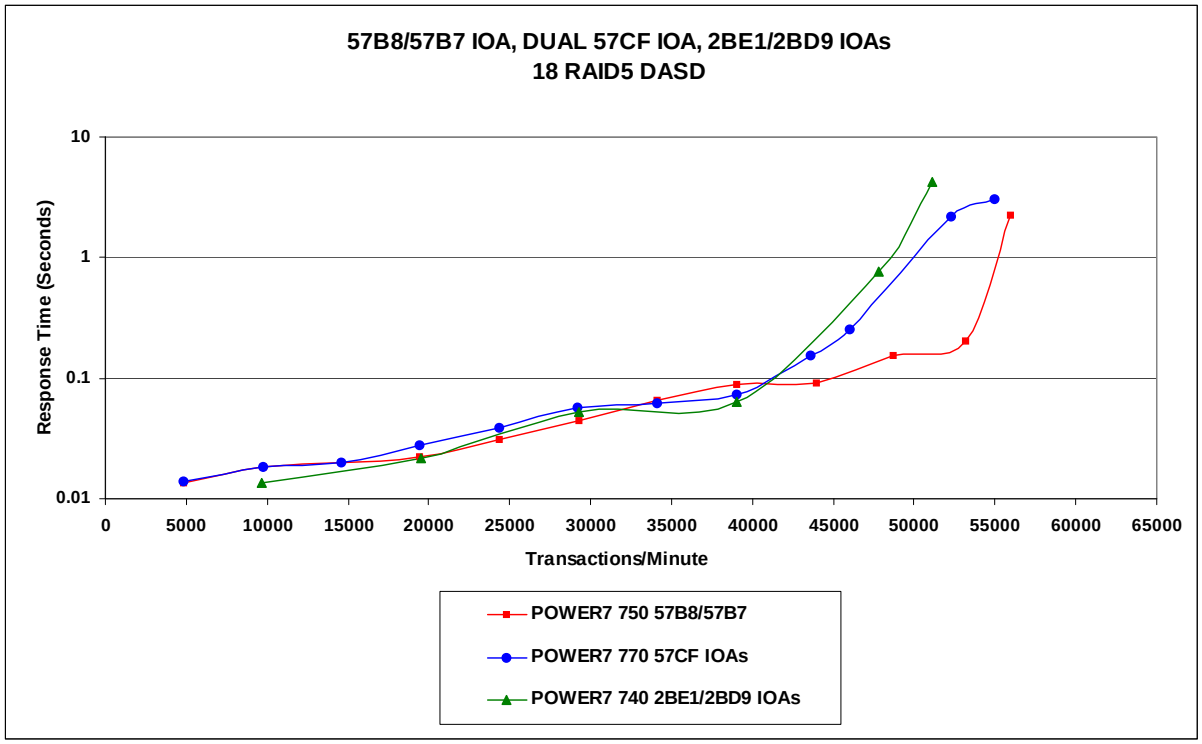


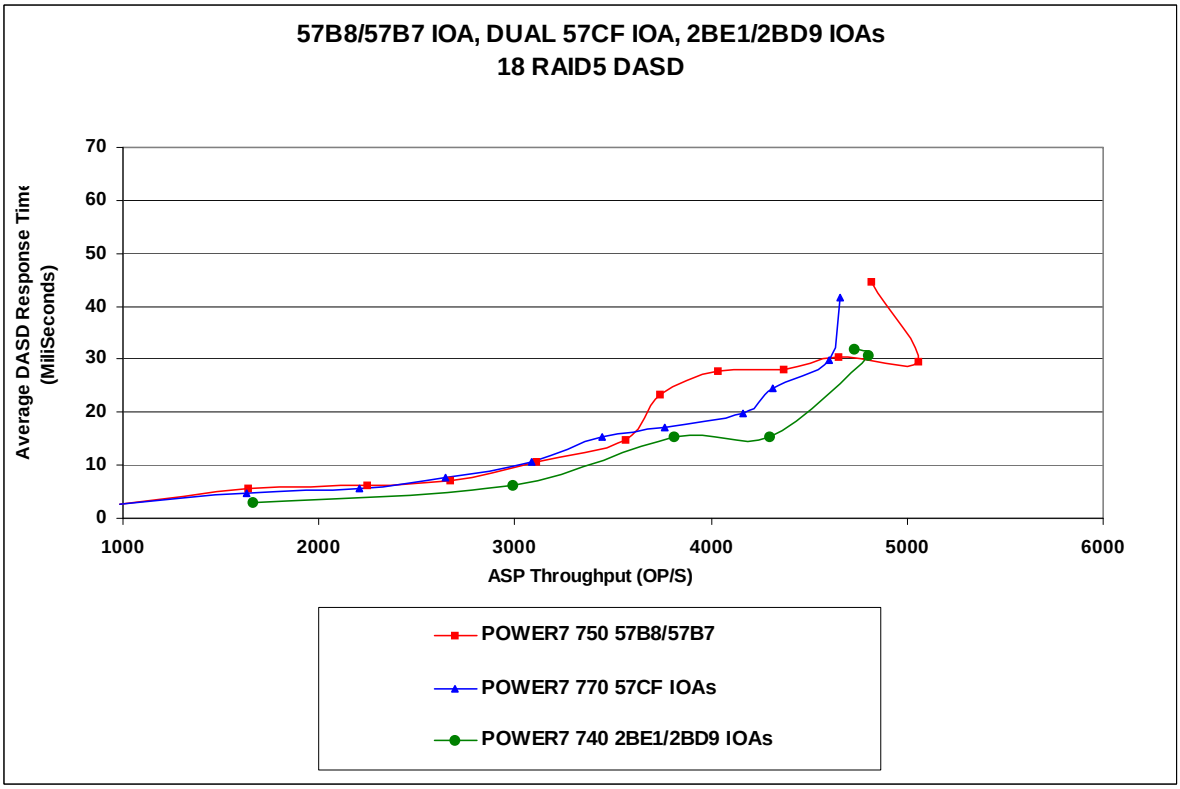






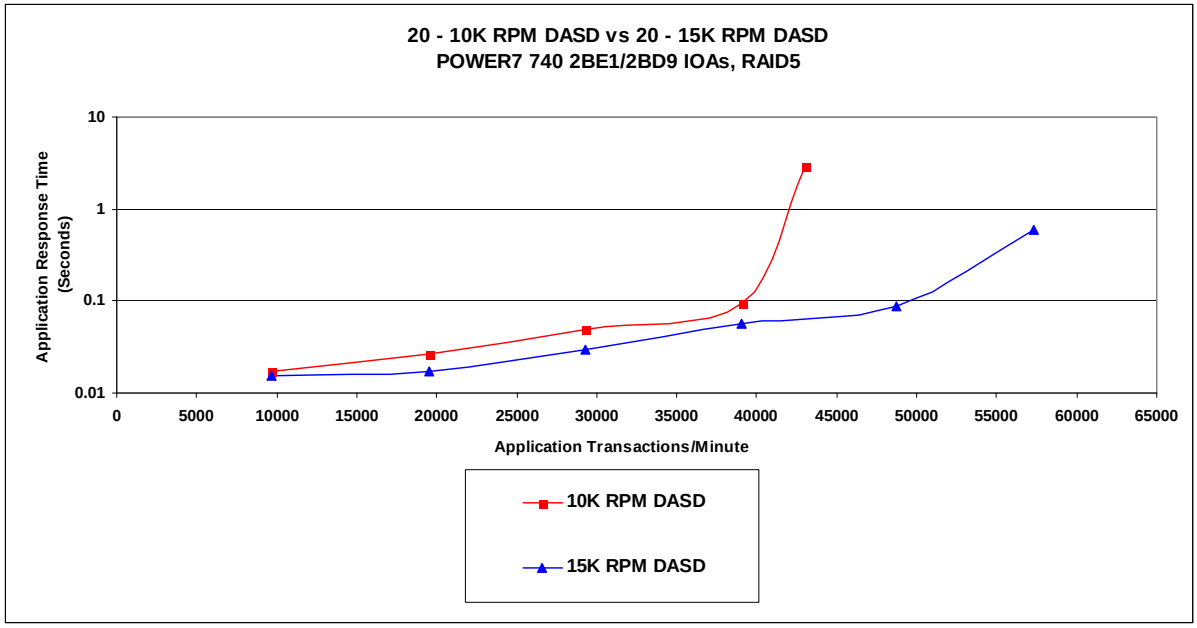
4.1.21 POWER7 740 2BE1/2BD9 IOA and storage expansion port.

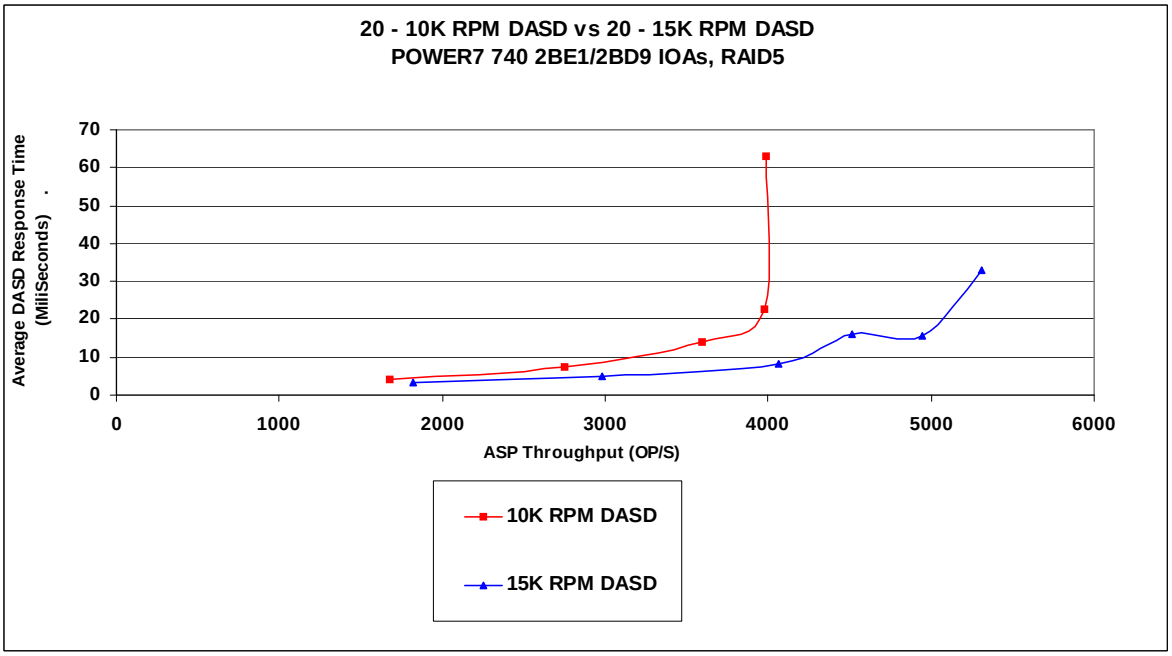




4.1.22 10K RPM DASD vs 15K RPM DASD

The 15k rpm DASD have a clear advantage when the IO is driven very hard. In an enviroment where the IO ops will remain lower the 10k rpm DASD can offer fair performance at a low price.

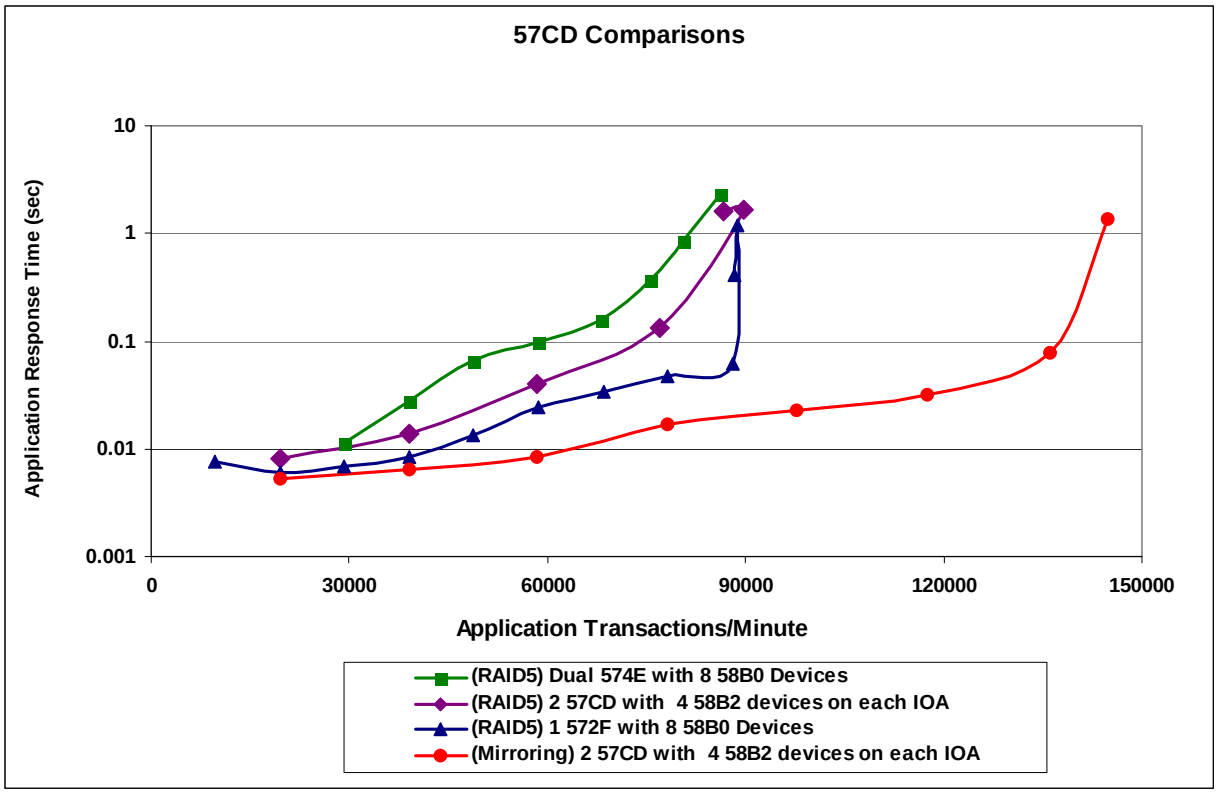


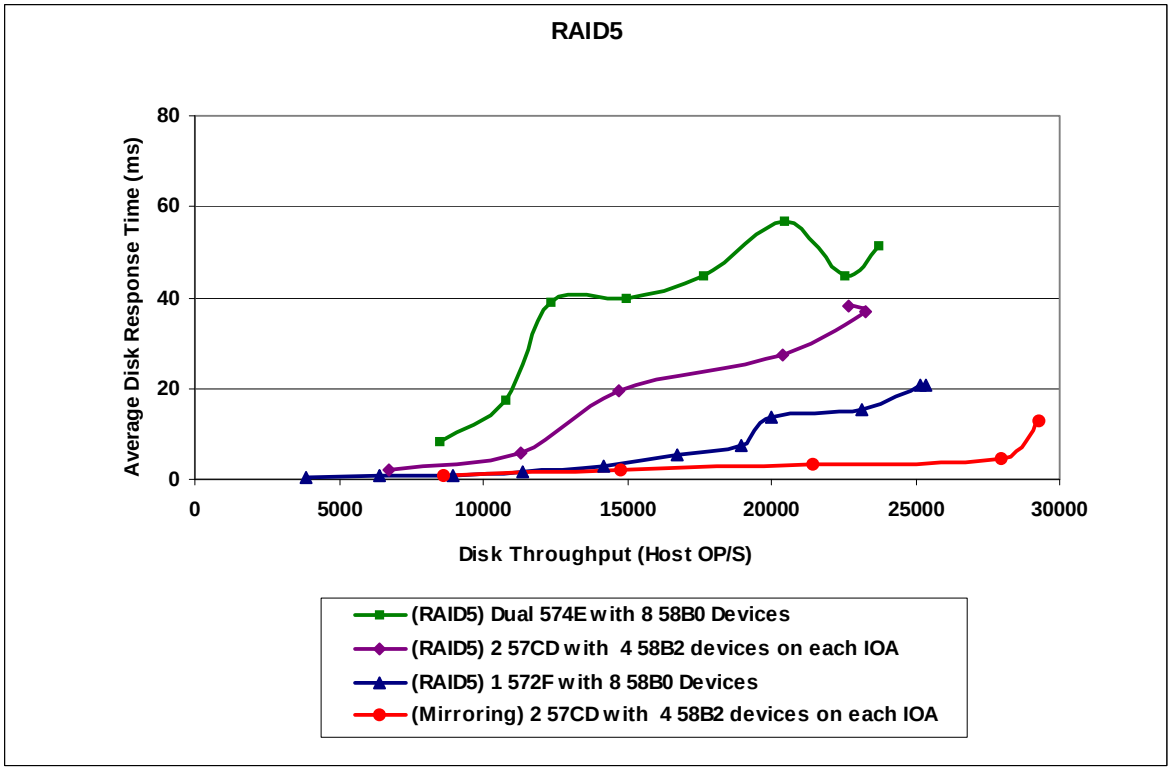


4.2 57CD IOA and 58B2 Devices

The 57CD IOA allows 1, 2 or 4 58B2 SSD devices per IOA and occupies two PCIe slots. If a customer has open PCIe slots the IOA is an opportunity to add storage without changing the current machine space. RAID5, RAID6 and Mirroring are supported, but because there is no write cache on the IOA our Rochester lab performance measurements show the value of mirroring when looking at the performance characteristics of the IOA and devices. The added value of mirroring is that concurrent repair is possible if a failure occurs.

The following charts compare RAID5 configurations from the current 58B0 SSD devices to the new 58B2 devices as well as a mirroring comparison.



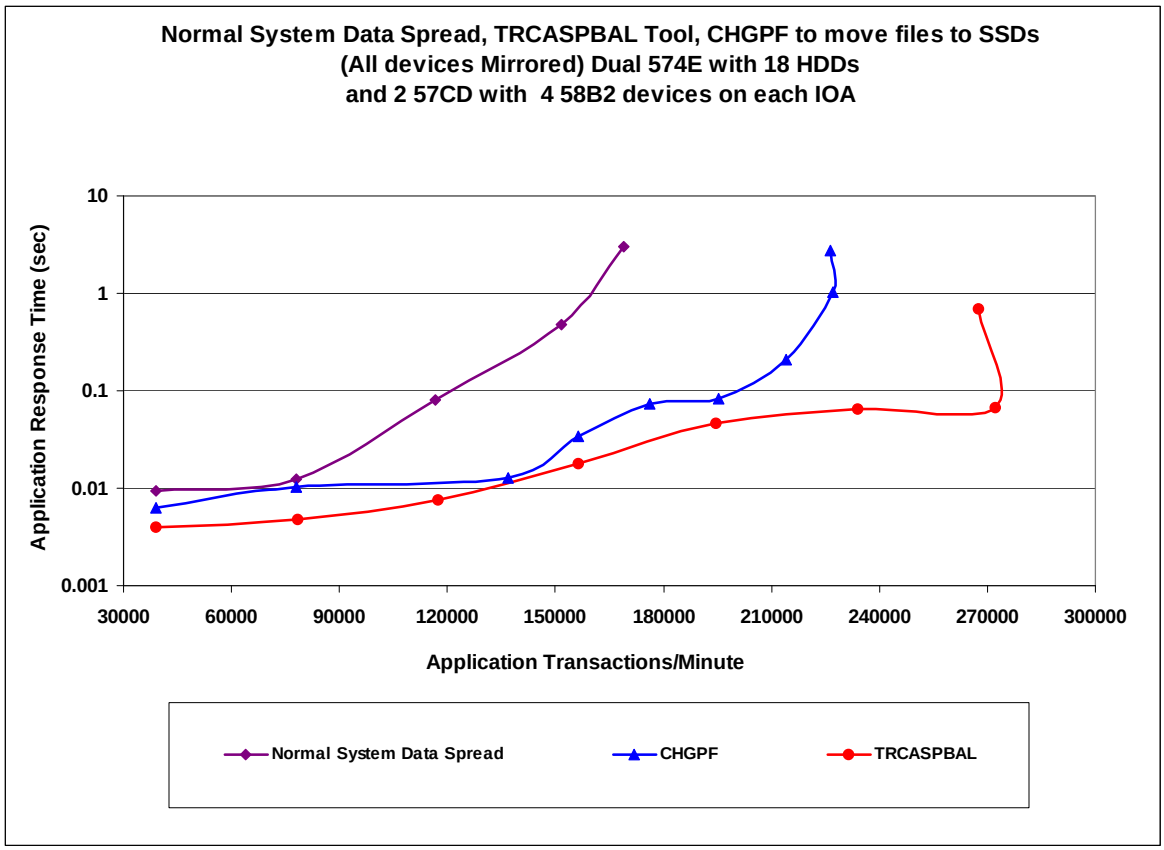


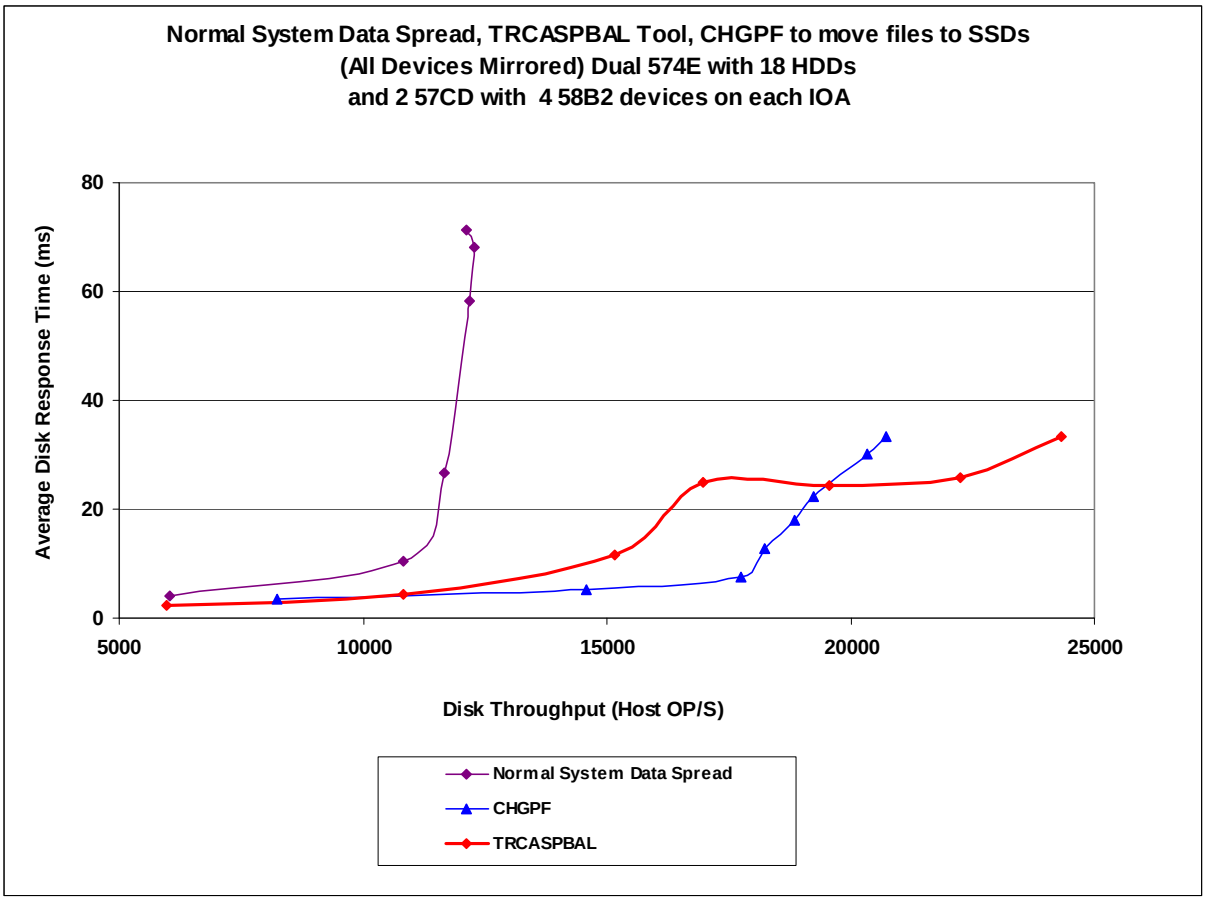
4.3 Normal System Data Spread, ASP Balancing Tool (TRCASPBAL), DB2 Media Preference Flag (CHGPF) to move files to SSDs

The Rochester lab ran some comparison workload runs to look at the performance characteristics when 2 57CD IOAs with 8 58B2 SSD devices were added to an ASP where our database was built to spread the data evenly across the HDDs and SSDs. After a couple baseline runs were complete we used the ASP balancer to trace and move the hot data to the SSDs. The database was then rebuilt without the SSDs in the configuration. The SSDs were added into the ASP and the DB2 Media Preference flag was used to move some specific files to our SSDs using the CHGPF command. PEX data along with Collection Services data was used to find the hot files that had a high majority of disk read requests.

The following charts show the results of those experiments. Our experiments showed better results from the ASP Balancing tool, but if we had worked more with the files in our workload we probably could have gotten better results from the CHGPF command also. The use of SSDs along with the movement of the hot data offered significantly better results.

If a customer knows their data files well the CHGPF command can be a simple way to get the most out of your SSDs if you aren't sure about your workload files then the ASP balancing tool offers a way to achieve the movement of your hot data onto your SSDs.



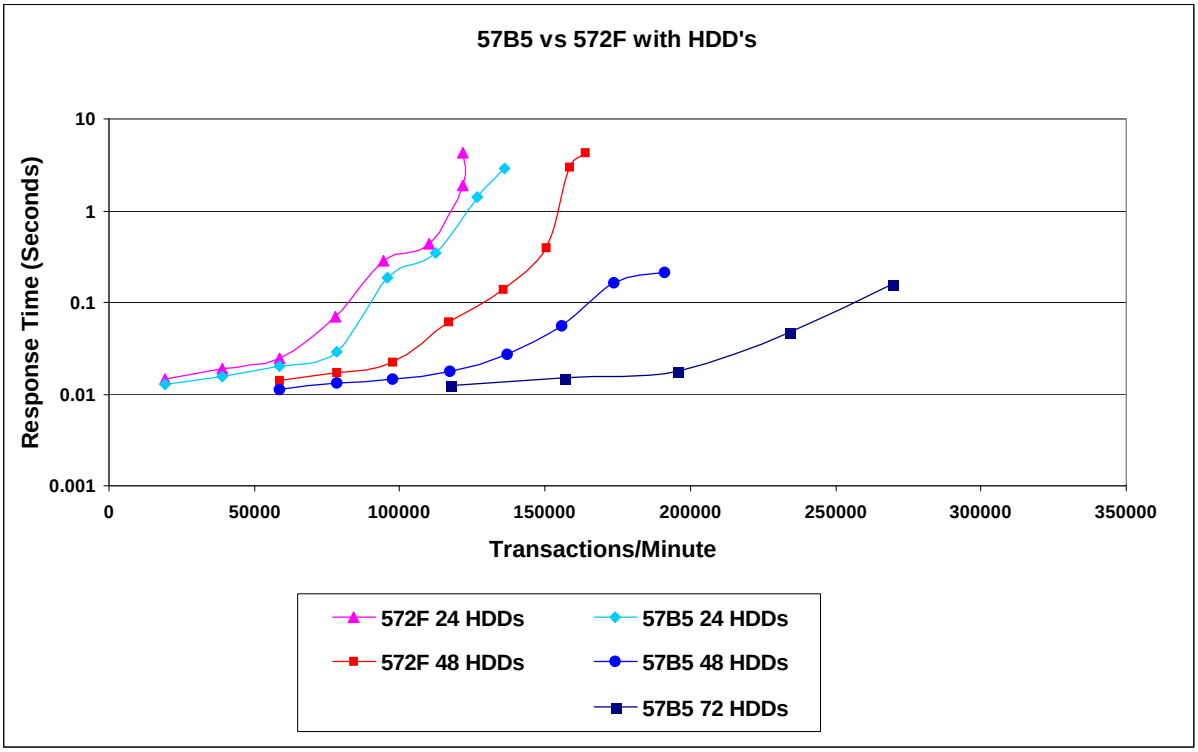


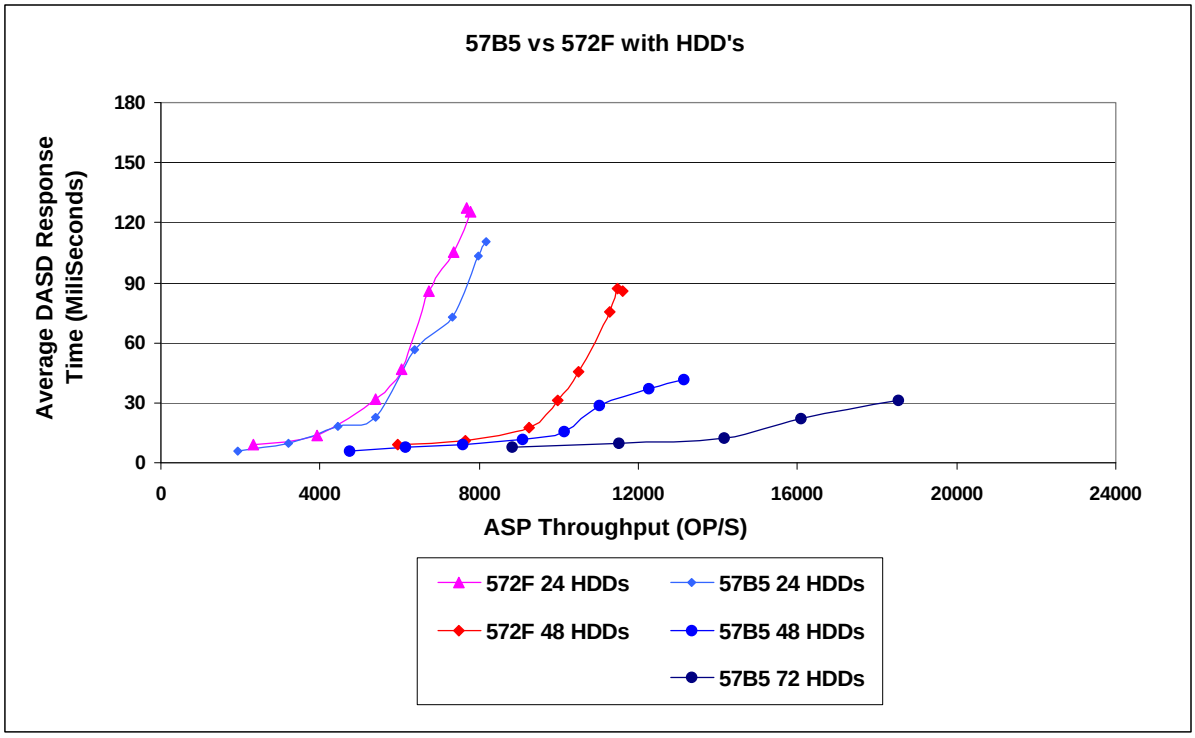
4.4 57B5 IOAs

The 57B5 is a new IOA, used in pairs to create a more flexible storage subsystem than previous storage IOAs. The 57B5 supports 5886 and 5887 drawers which can contain previous devices as well as the new devices being announced at the time this IOA is available. The following pages offer information on some of the testing done in the IBM lab, offering some comparison with current devices.

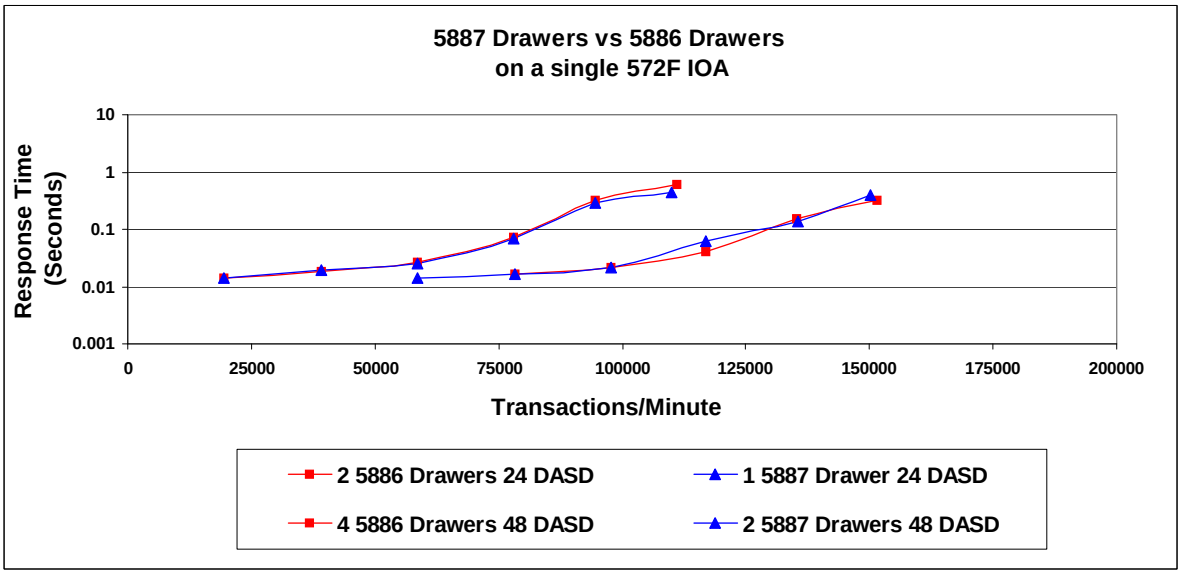
4.4.1 57B5 and 572F IOAs and HDDs in 5886 and 5887 Drawers

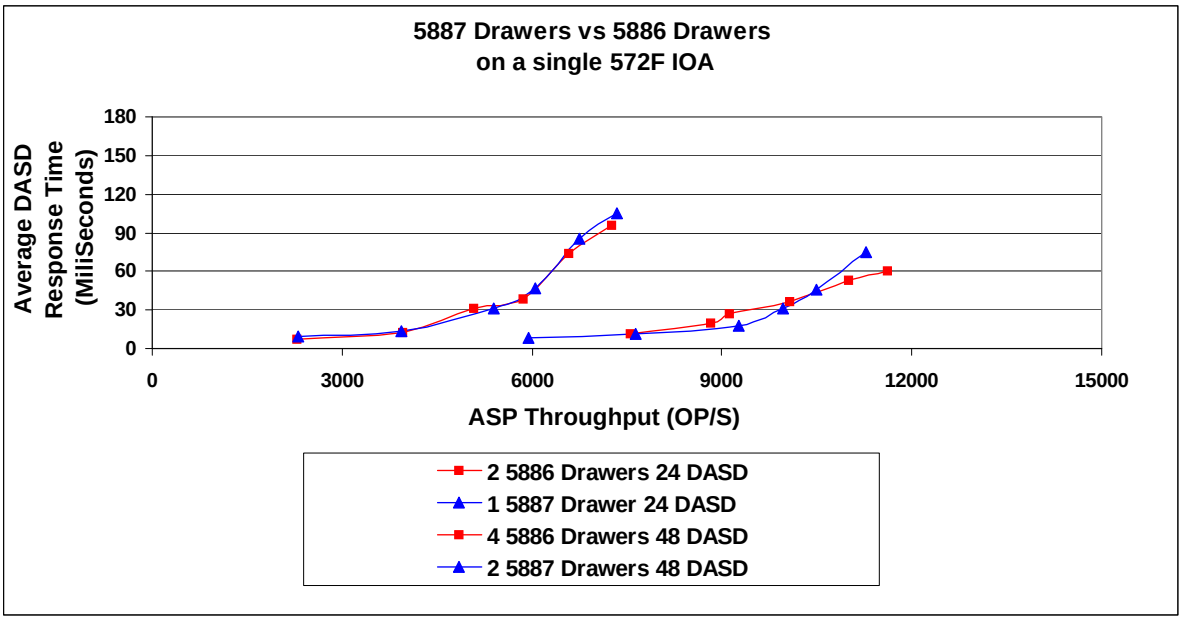
With HDDs the 57B5 shows a slight advantage over the 572F, however it needs to be noted that the 57B5 is used in pairs and takes advantage of the two active paths, where as the 572F is generally found as a stand alone IOA using only 1 path. The big item in the first chart is that although the 572F can support up to 60 HDDs after about 36 HDDs the performance is the same all the way up to 60 HDDs. This becomes more of a capacity expansion without any performance benefit from those devices. Where as the 57B5 IOA scales up to its maximum number of 72 HDDs.



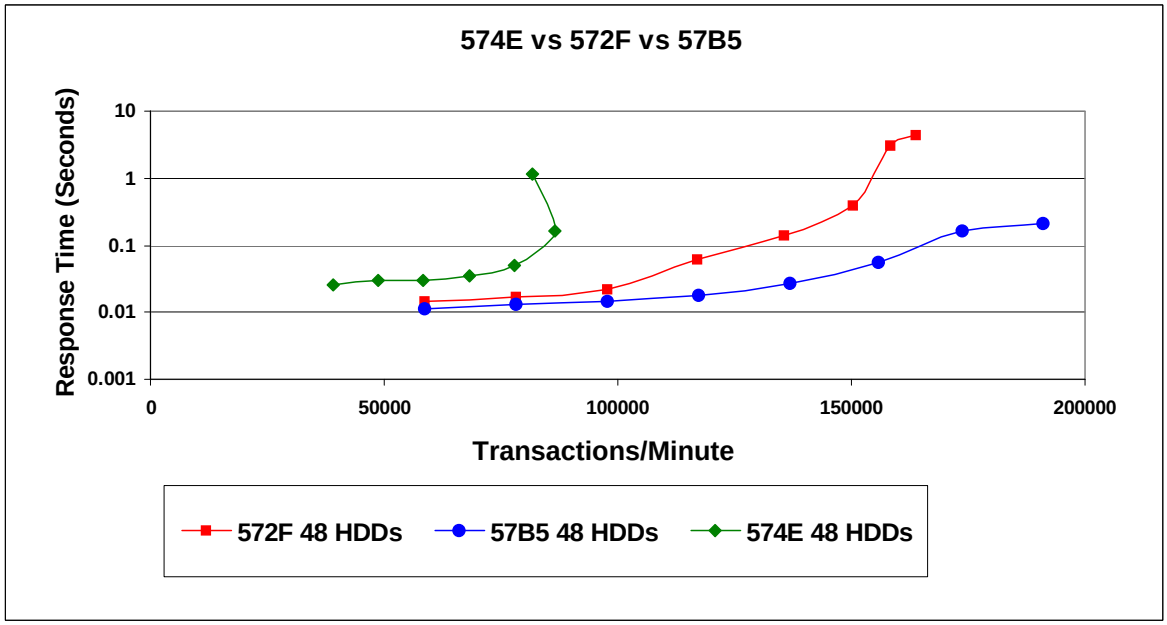


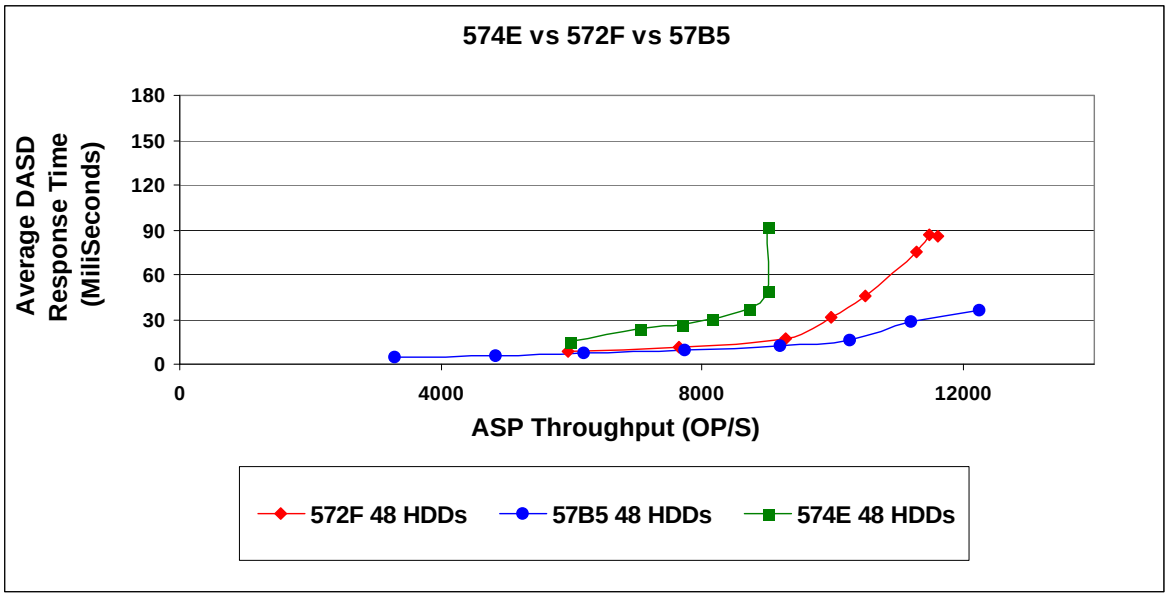
The 5886 and 5887 Drawers are supported on both the 572F and 57B5 IOAs. For the HDDs in the testing there was no difference in the workload based on the drawer we chose. There are differences in these as the 5886 drawer holds only 12 devices and the 5887 holds 24 devices in the same space. The 5586 drawers would also have to be chained together to get to the maximum number of devices. It does offer the option for those customers who might have the 5886 drawers currently.



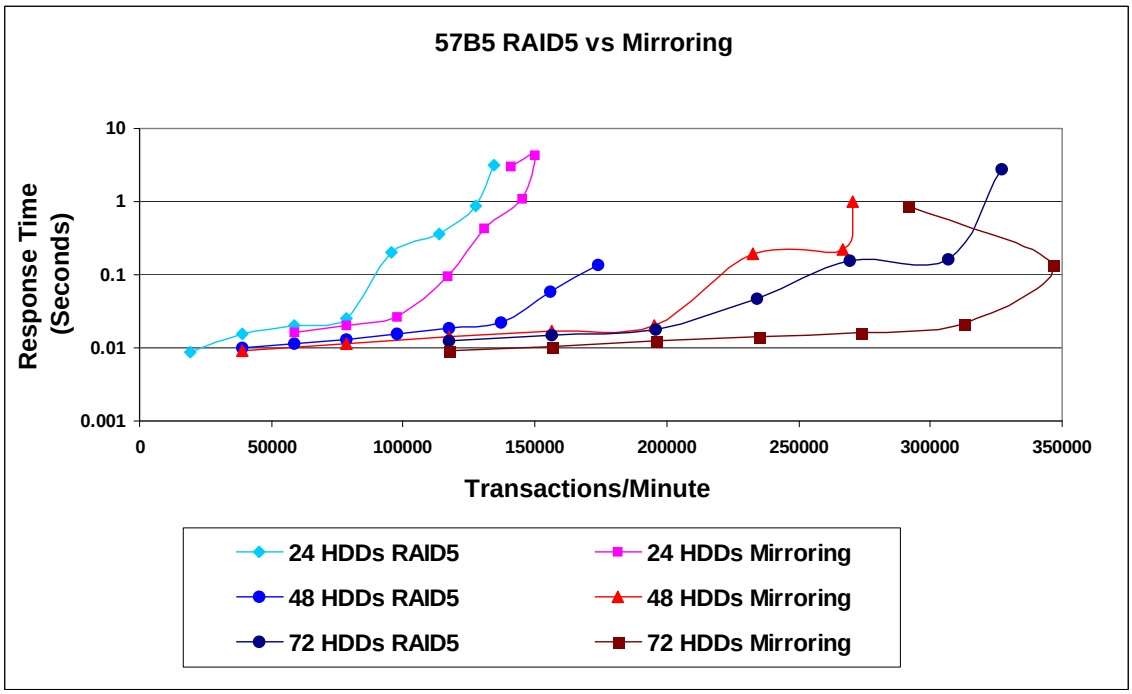


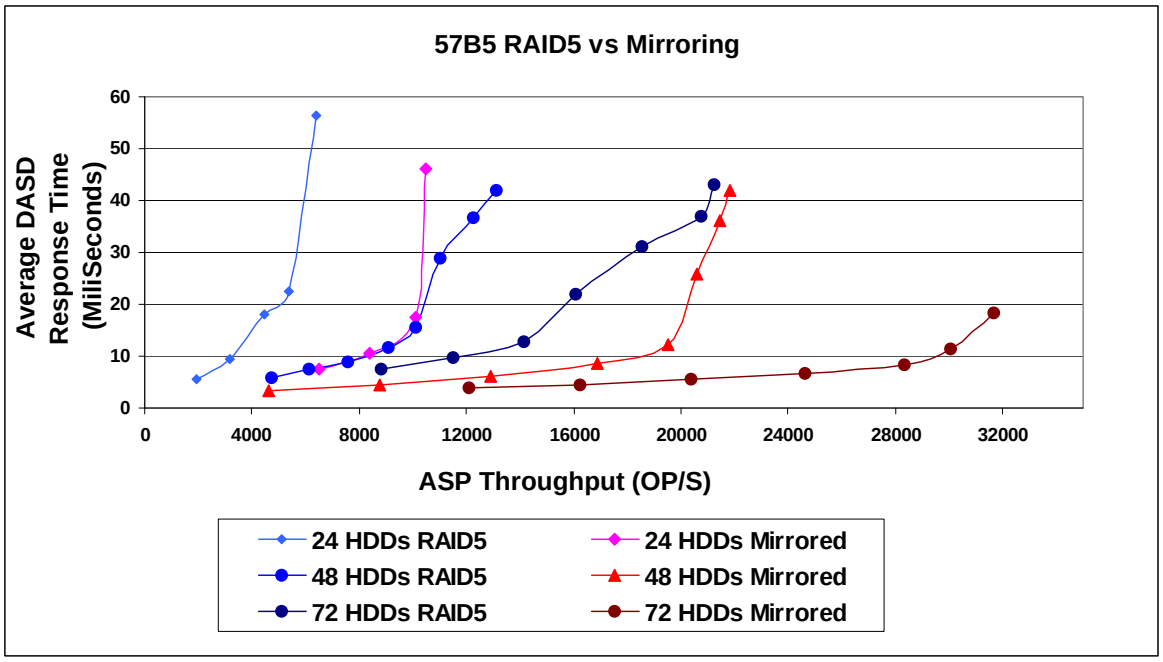
The other IOA available that we wanted a comparison point to is the 574E IOA this IOA is also used in pairs has a much smaller cache and has a lower cost. All factors with looking at which environment best fits the customer needs.





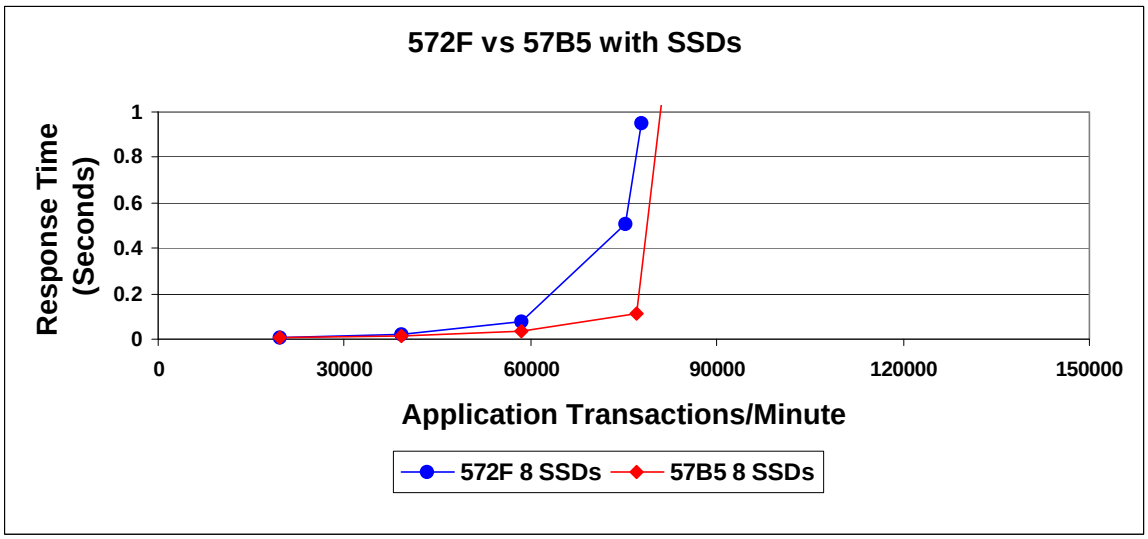
The next factor is RAID5 vs Mirroring, Mirroring takes more resource but is generally though of as a stronger protection scheme. The mirrored environment offers an advantage in performance and the following charts help to show that advantage in our DASDIO workload environment.

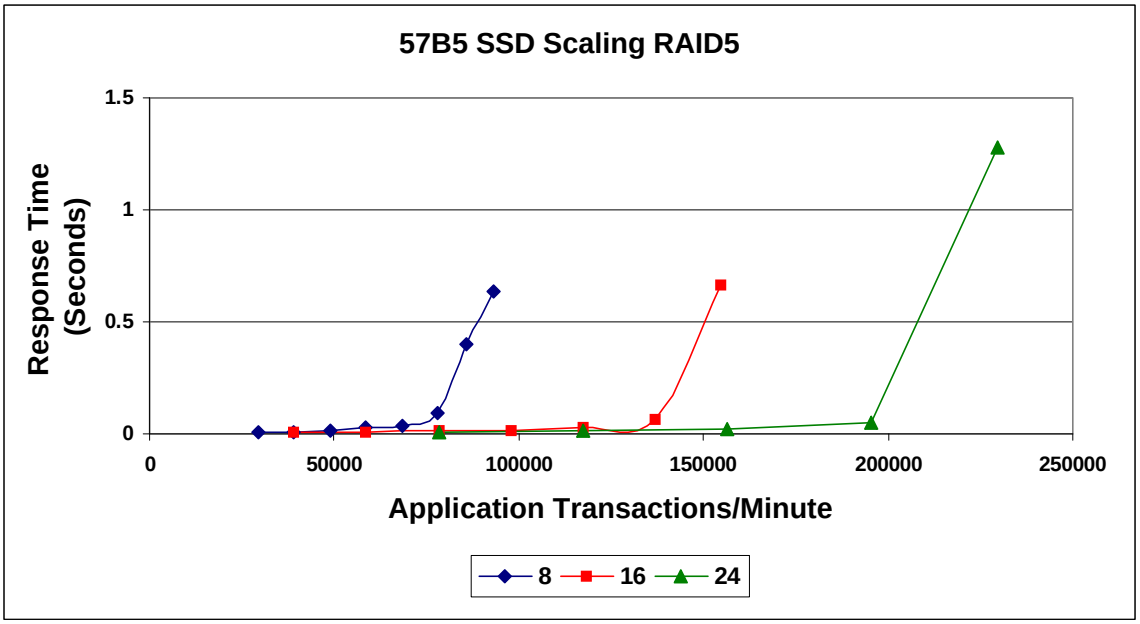


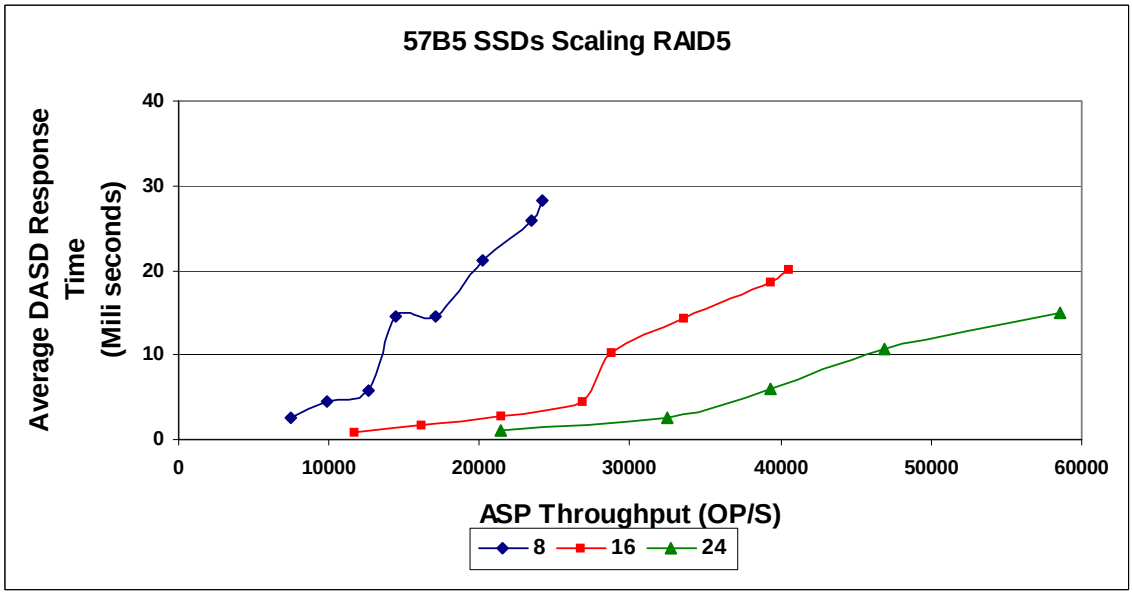


4.4.2 57B5 with SSDs

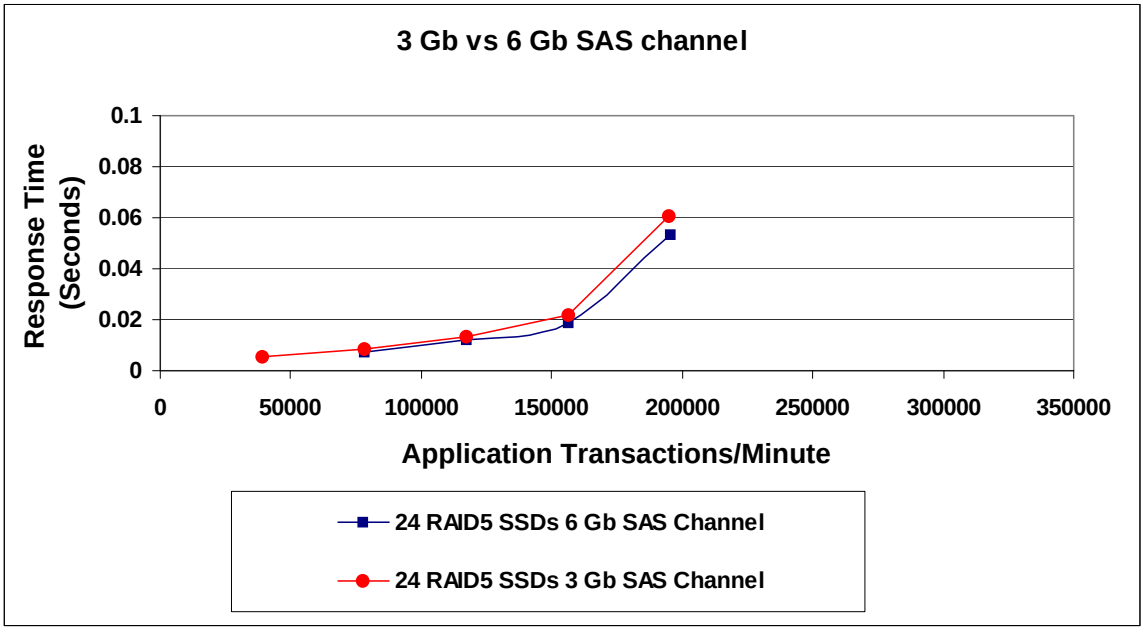
The first chart is a comparison with the 572F IOA. The 572F was limited to 8 SSDs and we had some scaling experiments which suggested 8 SSDs is where performance stopped. I used that in setting up the first charts. The 572F is a single IOA and the 57B5 is a pair of IOAs so it skews the comparison but the important factor here is that after 8 devices the 572F could no longer get more performance with the addition of devices. The 57B5 scales all the way up to 24 devices.

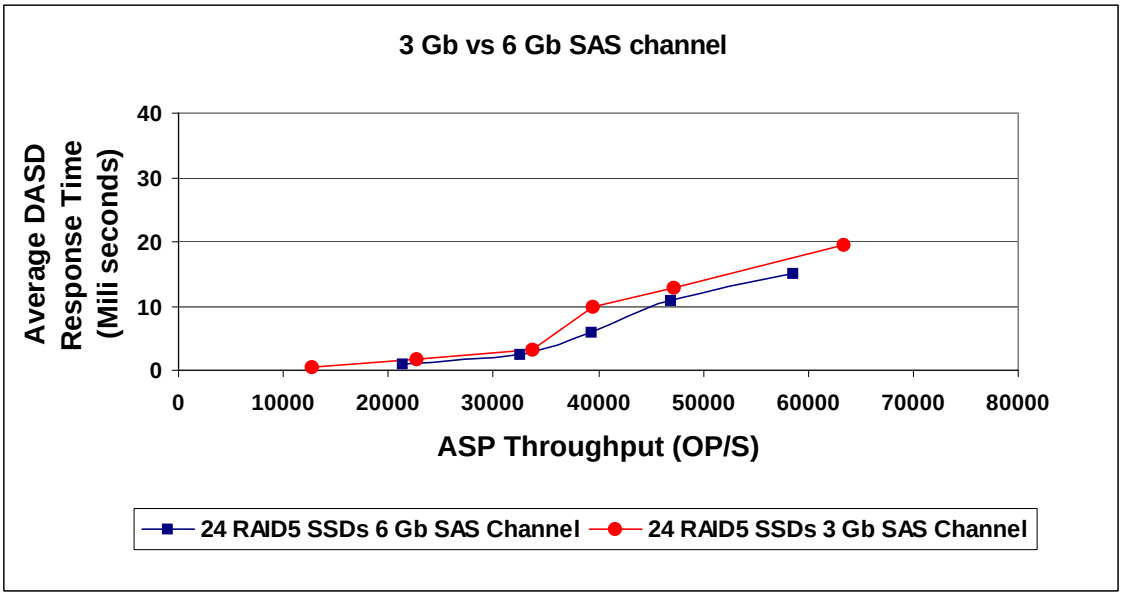




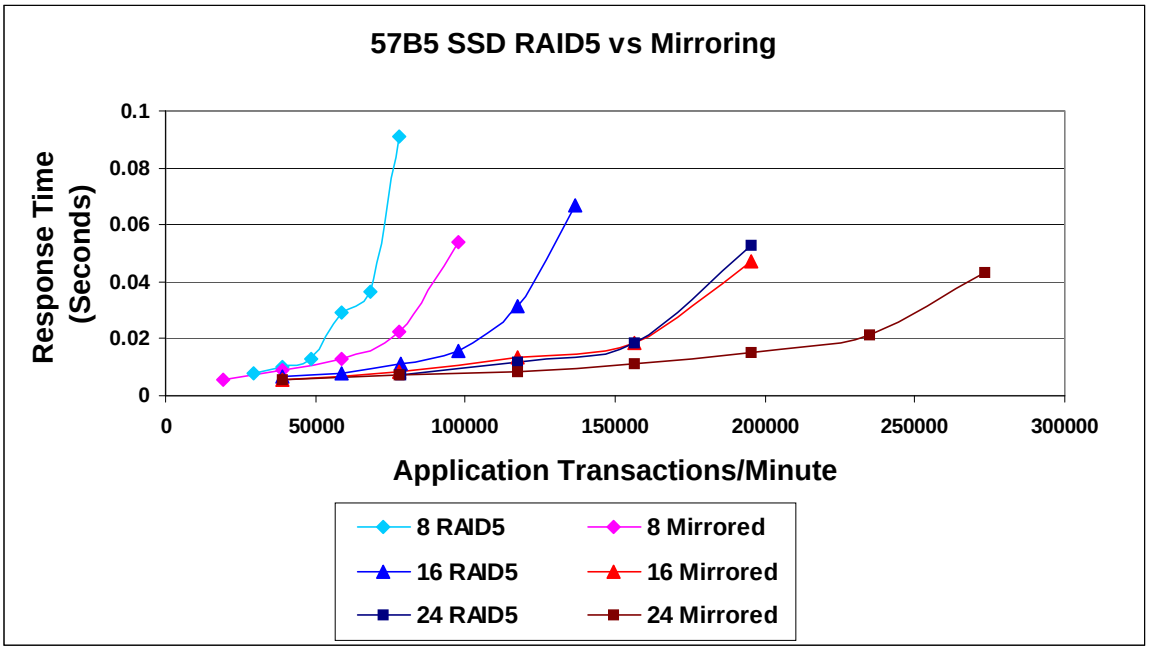


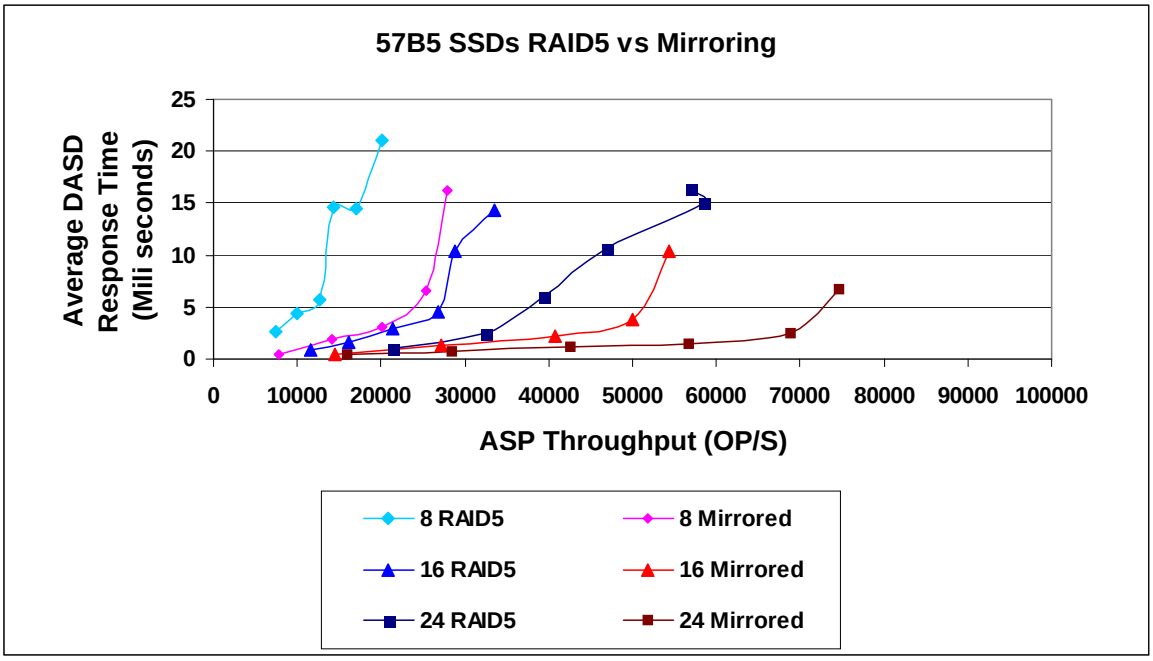
The 57B5 has a 6 Gb SAS connection and the 5887 drawers are also 6 Gb where as the 572F and the 5886 drawers used a 3 Gb SAS channel. In the basic experiments we did for the following charts there was no discernible difference with the DASDIO workload. This workload is a small block workload where the average kio is 8 to 12. Customers with large block workloads could see an advantage with the 6 Gb SAS channel.





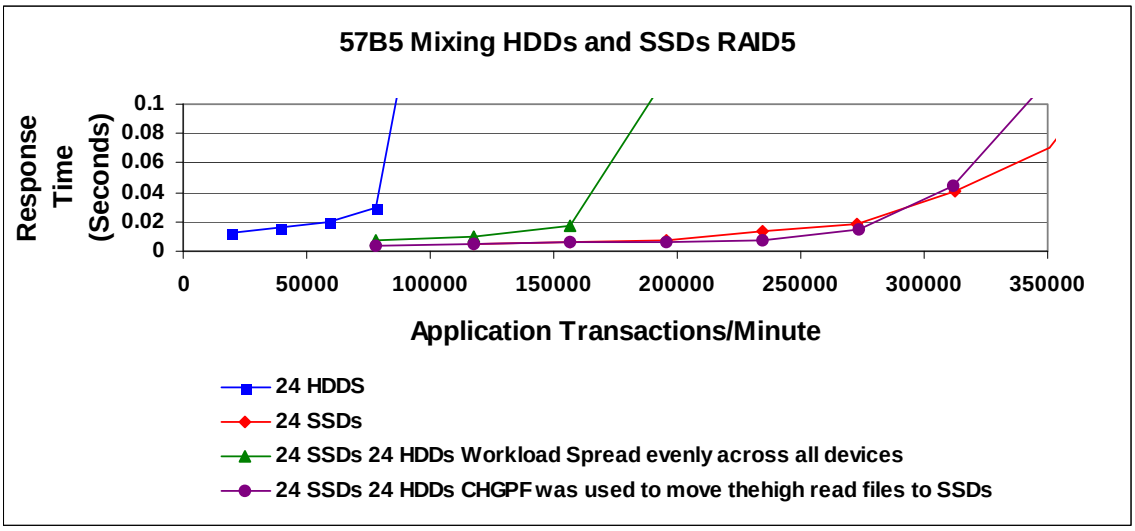
Mirroring is a bigger factor with the SSDs. In the following charts 16 Mirrored SSDs performed as well as 24 RAID5 SSDs.

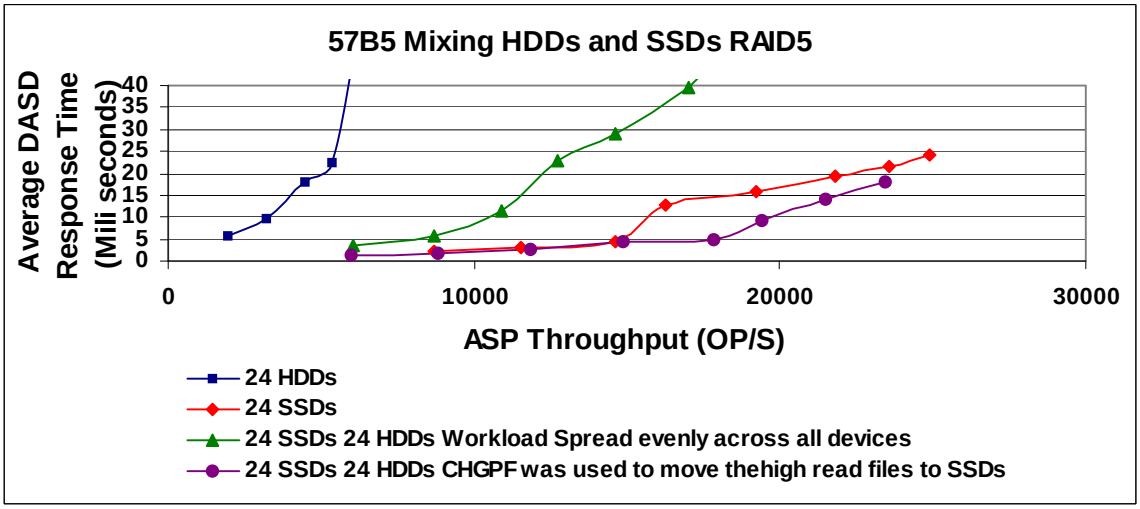




The 57B5 allows HDDs and SSDs to be on the same pair of IOAs. Two of the rules I will note here is that for IBMi the SSDs and HDDs cannot share the same port and SSDs are not supposed to be cabled off of the top ports on the IOA. A maximum of 24 SSDs along with 48 HDDs can be placed on a pair of 57B5 IOAs.

The most important thing to remember about this environment or any environment where the ASP is a mix of HDDs and SSDs is to balance your data. There are two methods to balance data on the SSDs so that the high read data is placed on the SSDs. The first is media preference if you know the files that are largely reads you can place them on the SSDs using the following. CHGPF FILE(Library/filename) UNIT(*SSD) or CHGLF FILE(library/filename) UNIT(*SSD). The second option is to use the TRCASPBAL tools. I knew which files I wanted to move so I chose CHGPF.





4.4.3 57B5 scaling with HDDs

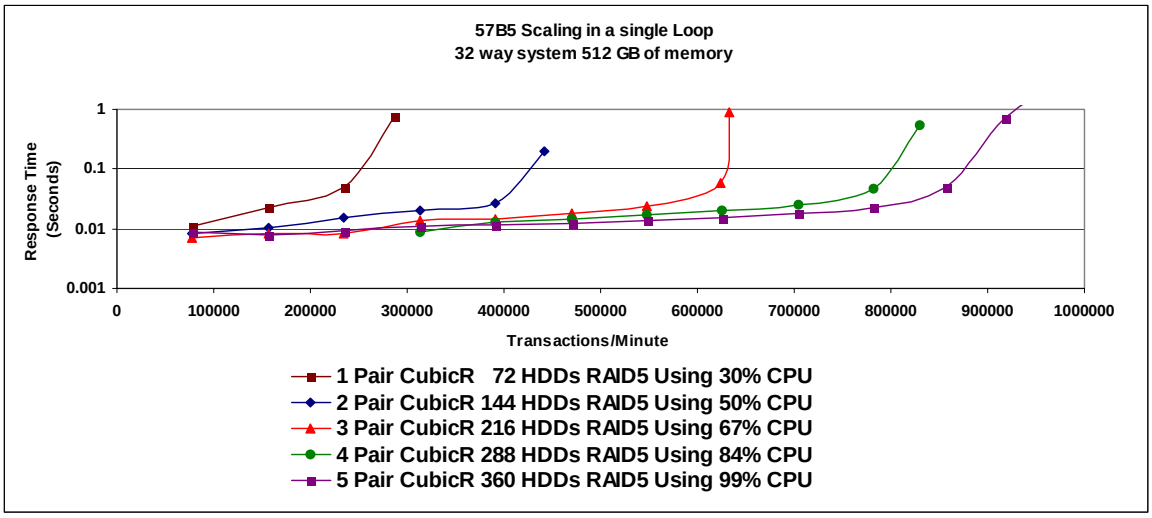
The following charts display a single HSL loop in which 1 to 5 pairs of 57B5 IOAs are added in and the DASDIO workload is plotted.

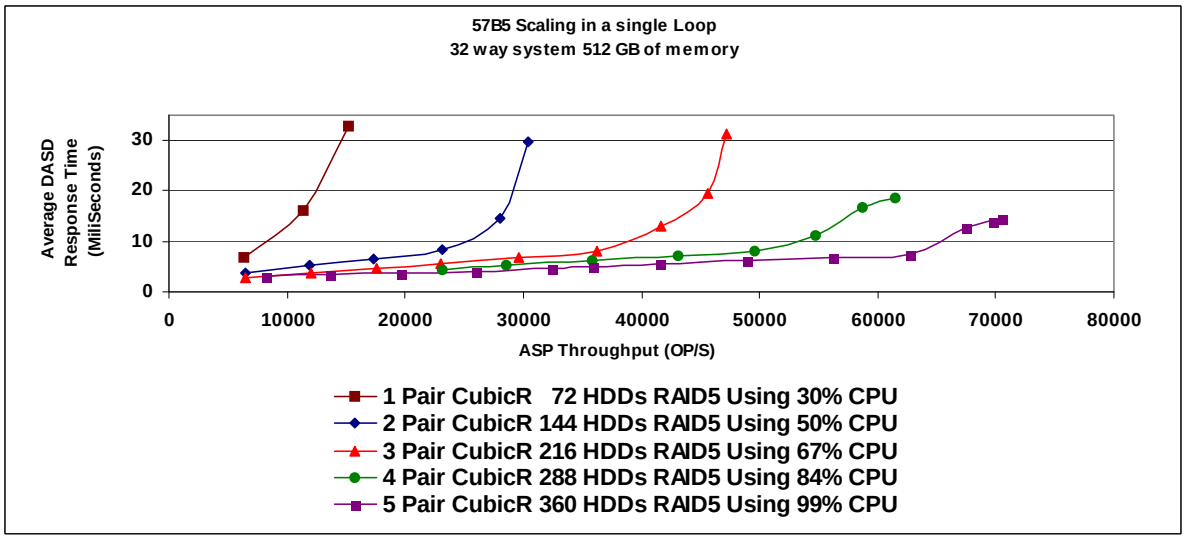
Each pair of IOAs has 72 HDDs in three XXXX drawers, all using RAID5 protection.

The system used was A 9179-MHD and a single 32 way partition with 512 GB of main storage.

At the point where the workload displayed a bottle neck the CPU usage was noted When the 5th pair of IOAs were added we noted the bottle neck was now system resources instead of IOA and storage resources.

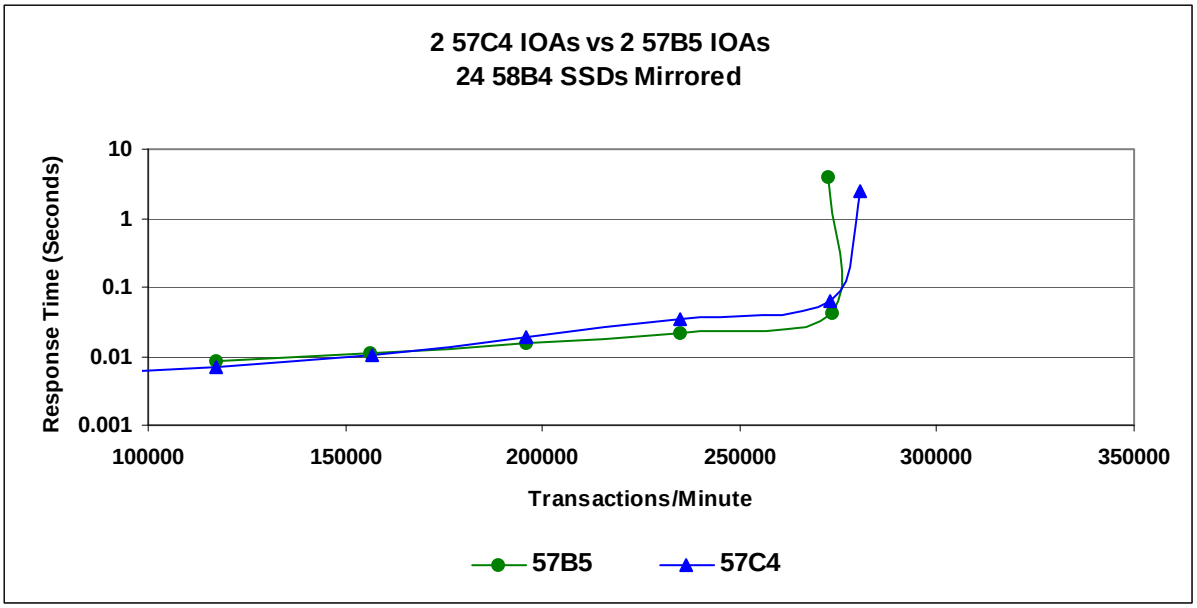
A total of 360 HDDs were in place when running all 5 pairs of IOAs. As more IOAs, HDDs, and loops can be added to a partition the value to performance diminishes and the additional HDDs would become more of an addition for storage capacity.

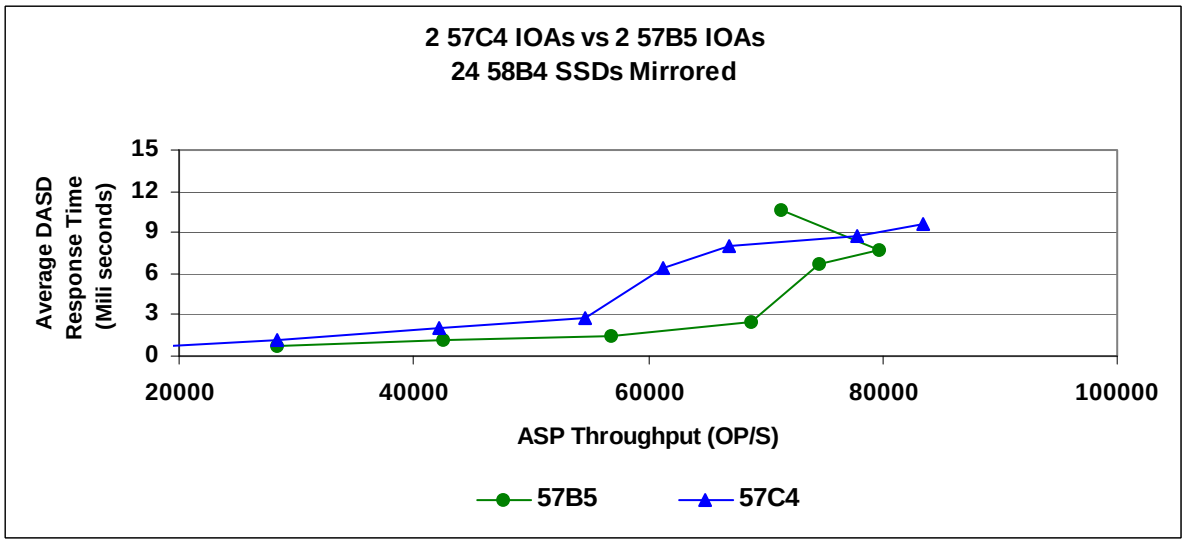




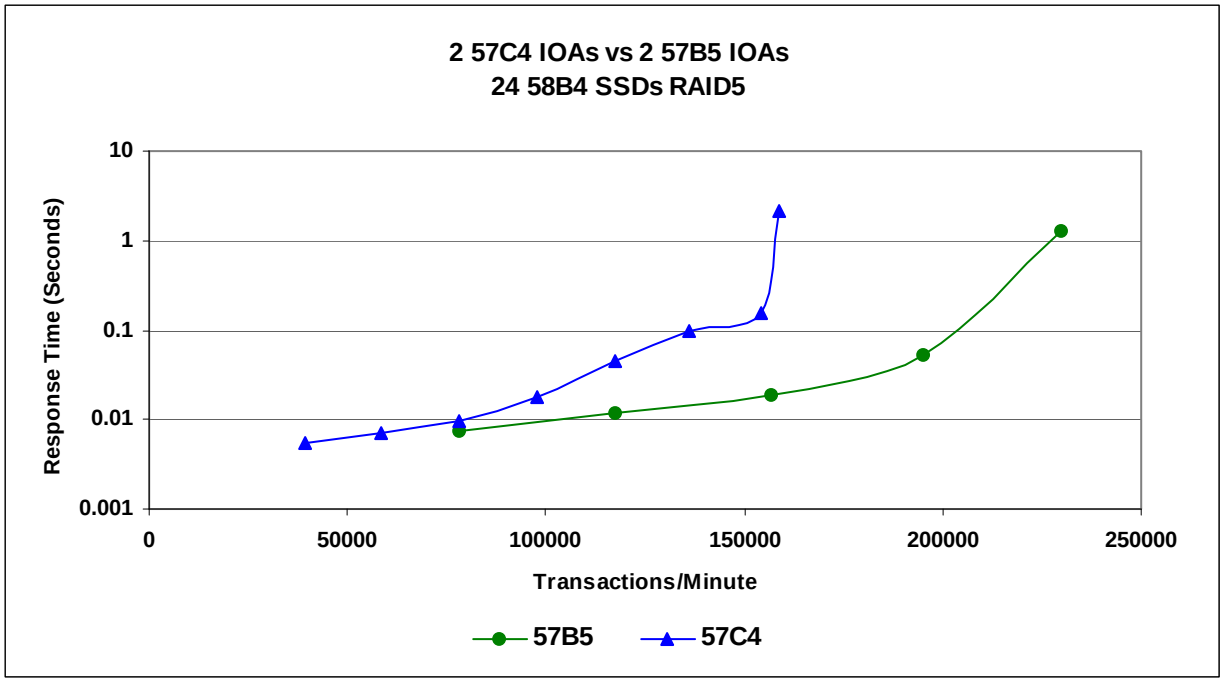
4.5 57C4 IOA

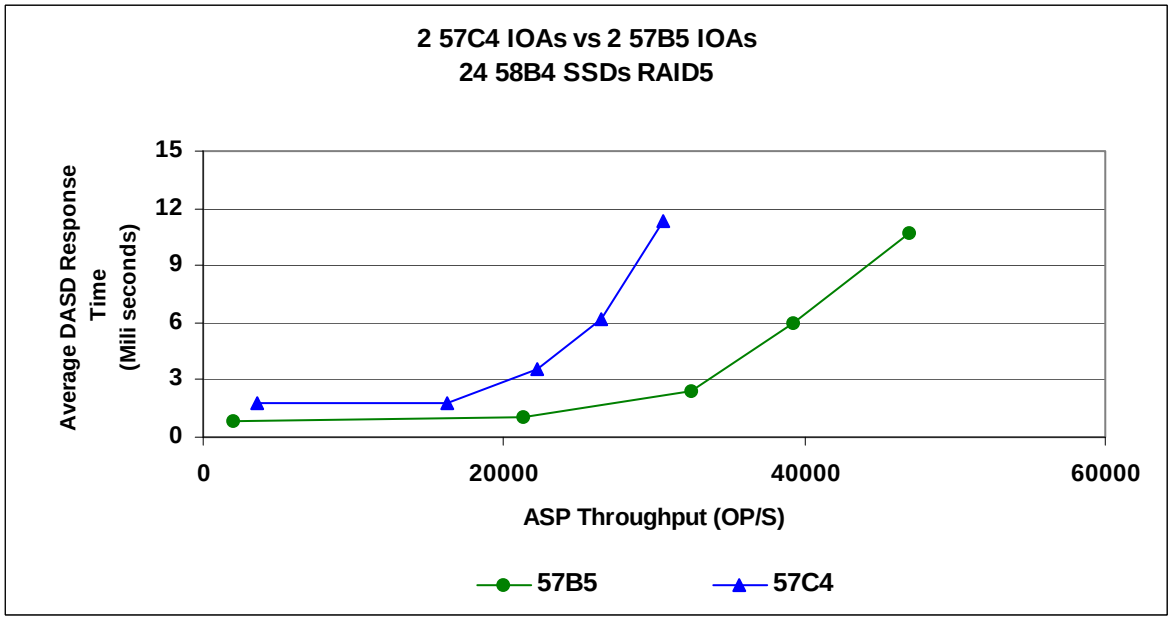
The 57C4 is a new IOA that can be used as a single IOA or in pairs but has no cache and only supports solid state devices. The following charts offer comparisons between a pair of 57C4's and a pair of 57B5's with the same 58B4 devices attached.
Using IBMi mirroring, the 57B5 and 57C4 IOA workload throughput is much the same. Without a cache though, as the IO's increase, the 57C4 response times rise quicker. The choice of the IOAs will need to be assessed for any given environment. If the IO's need high throughputs from each IOA card pair and the applications are response time sensitive, it may be more economical to purchase less of the 57B5s with cache than to purchase more of the 57C4s without cache.

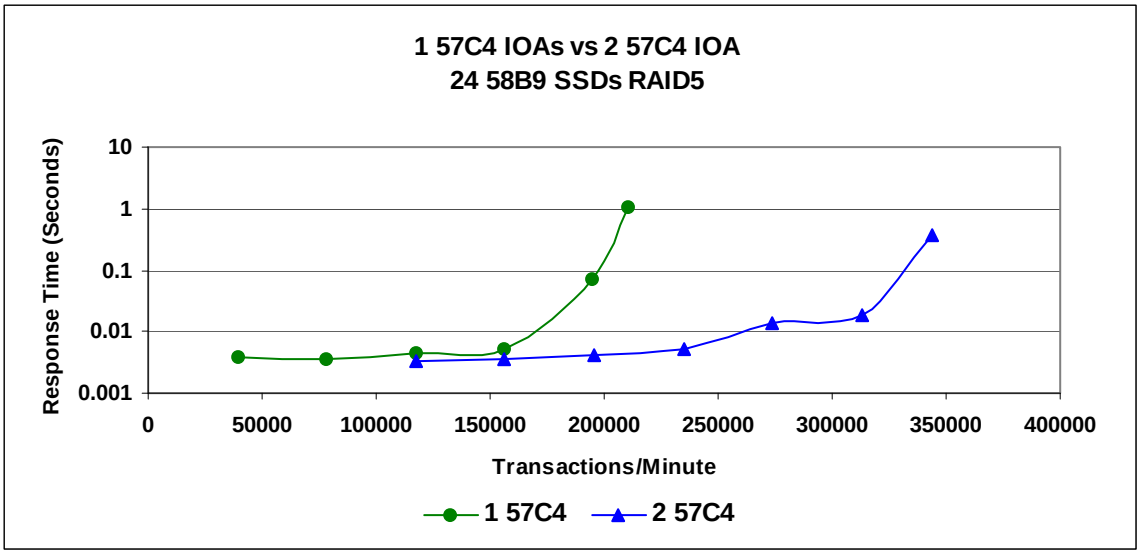


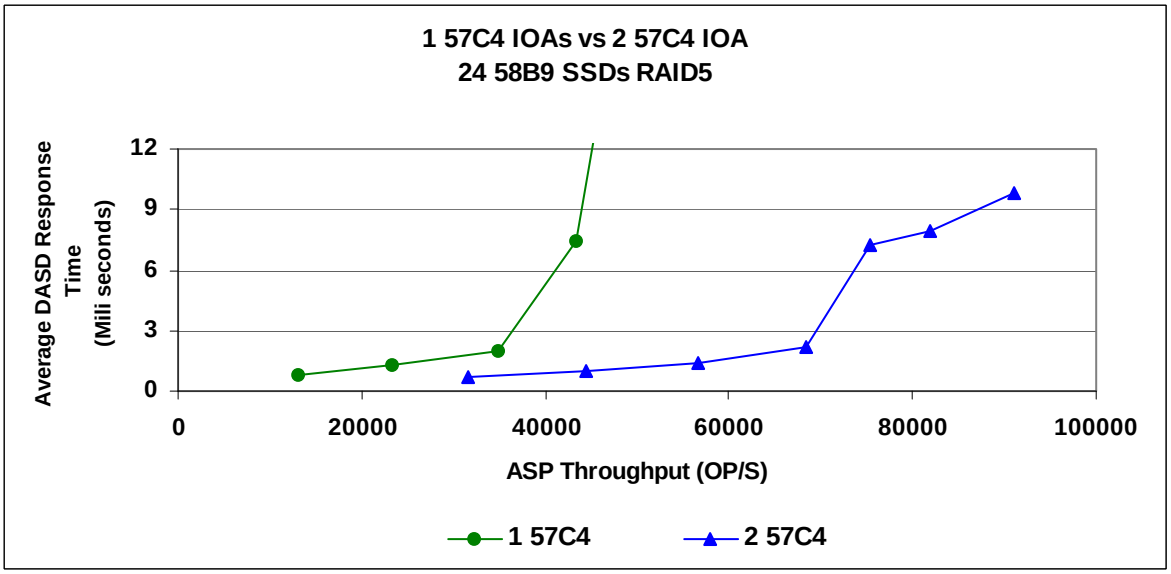


As seen in the charts below the lack of cache has a greater affect when using RAID5. The 57C4's response time effectiveness is similar to the 57B5's at lower IO rates, but Transactions/Minute for a constant number of users is aided by an adapter cache. So again, the cost of devices along with the throughput needs can make the 57C4 an attractive alternative.



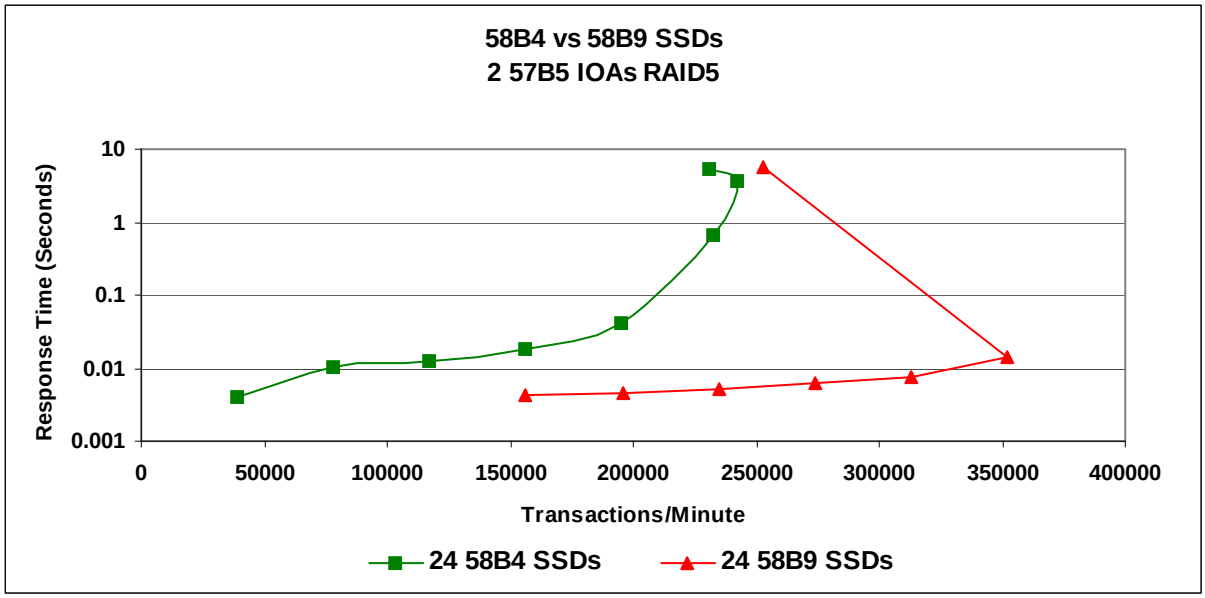


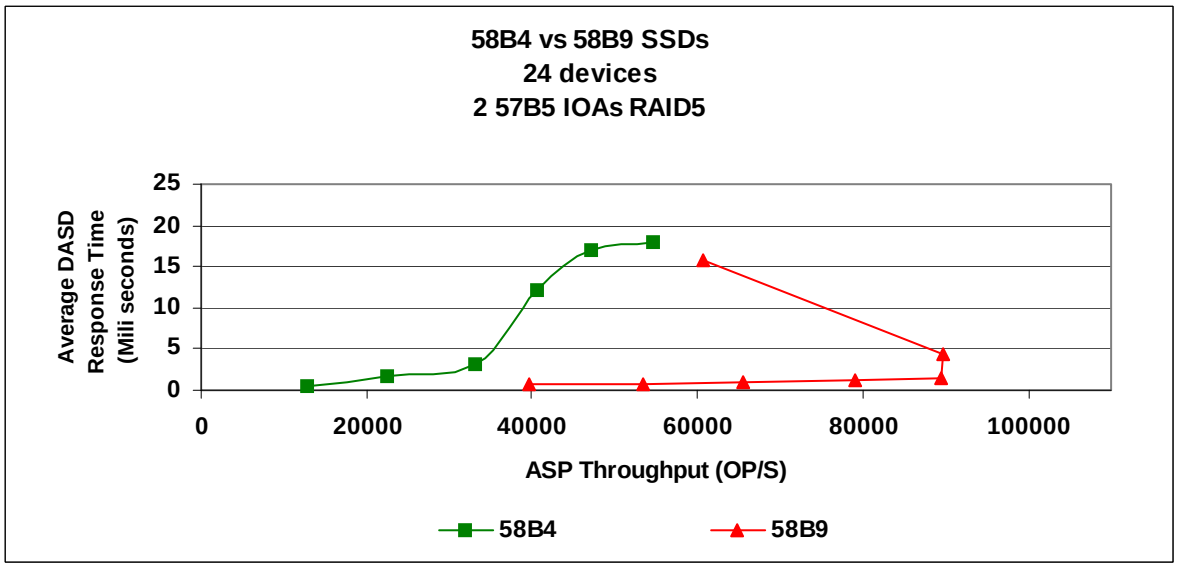




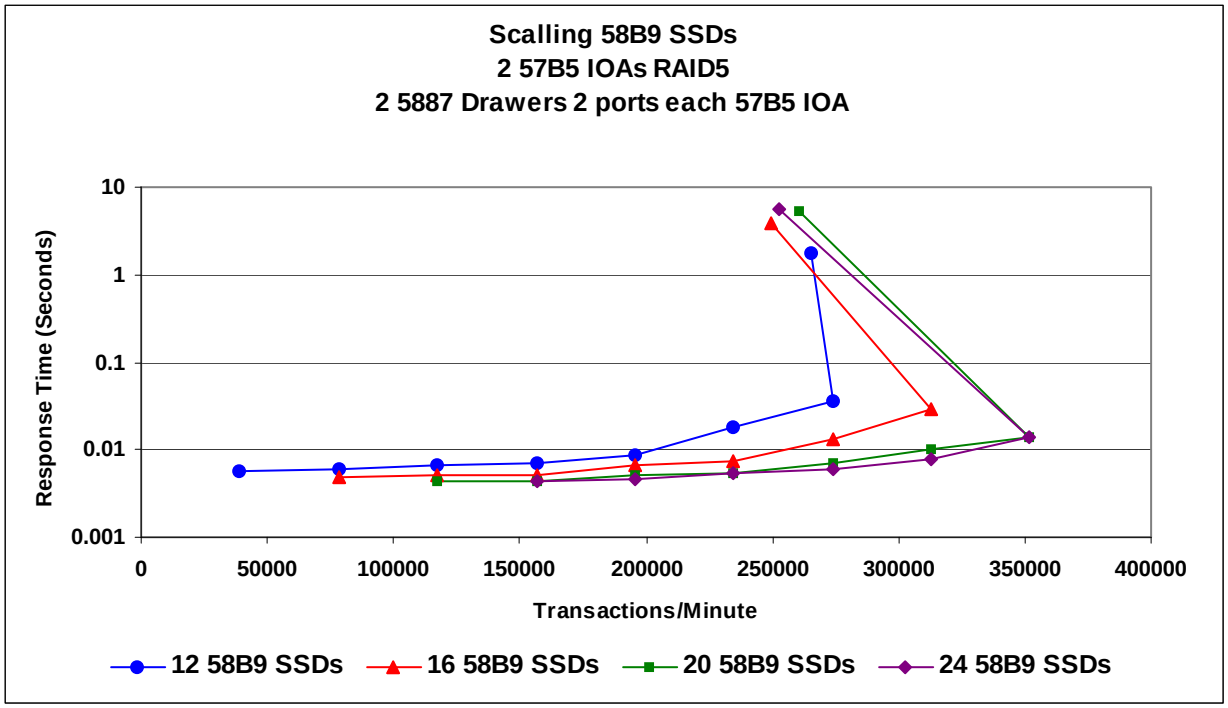
4.6 58B8/58B9 Solid State Devices

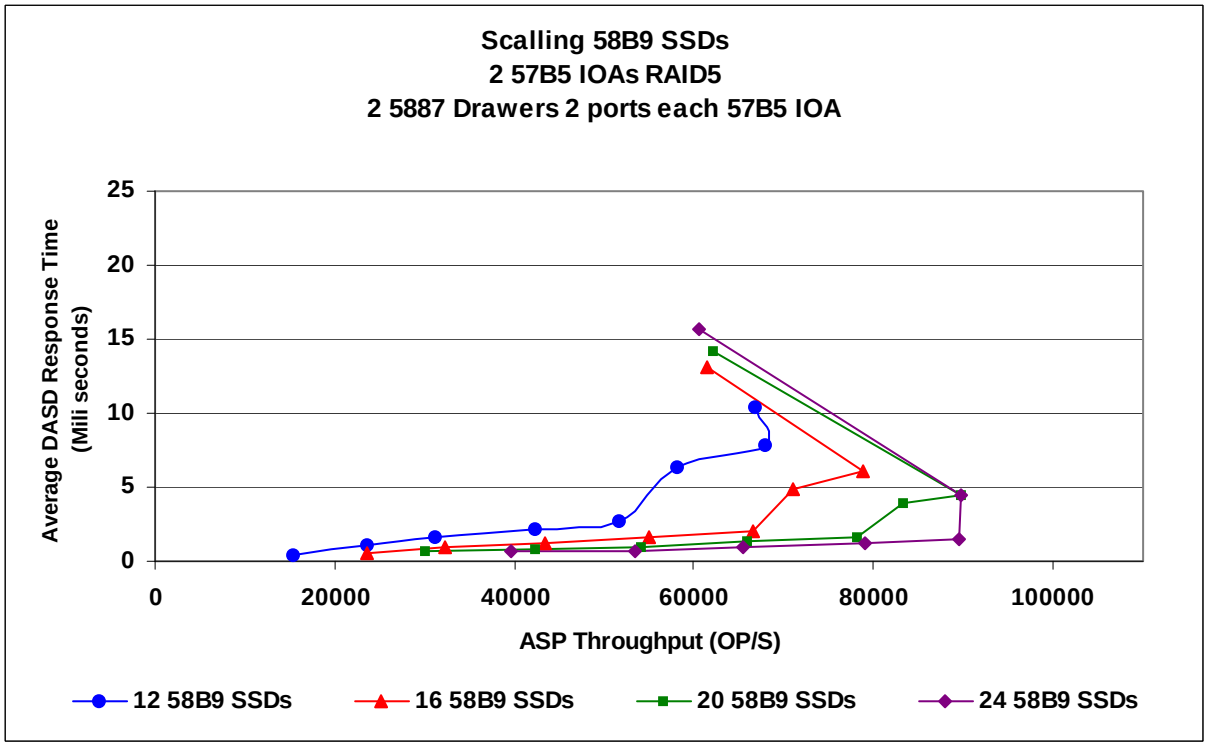
The 58B8/58B9 SSDs show a sizable increase in throughput over the predecessor 58B3/58B4, Enterprise MLC based SSDs. The following charts offer information showing that noticeable throughput improvement comparing Gen2 SAS bay carrier models. The DASDIO workloads used in the following charts have an average IO length of about 12KB.



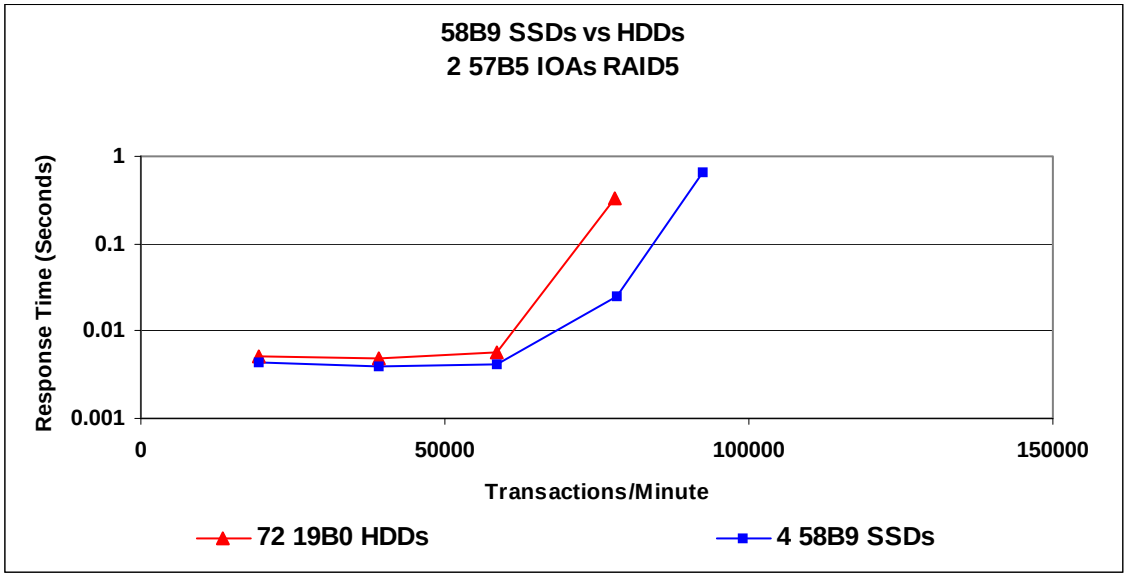


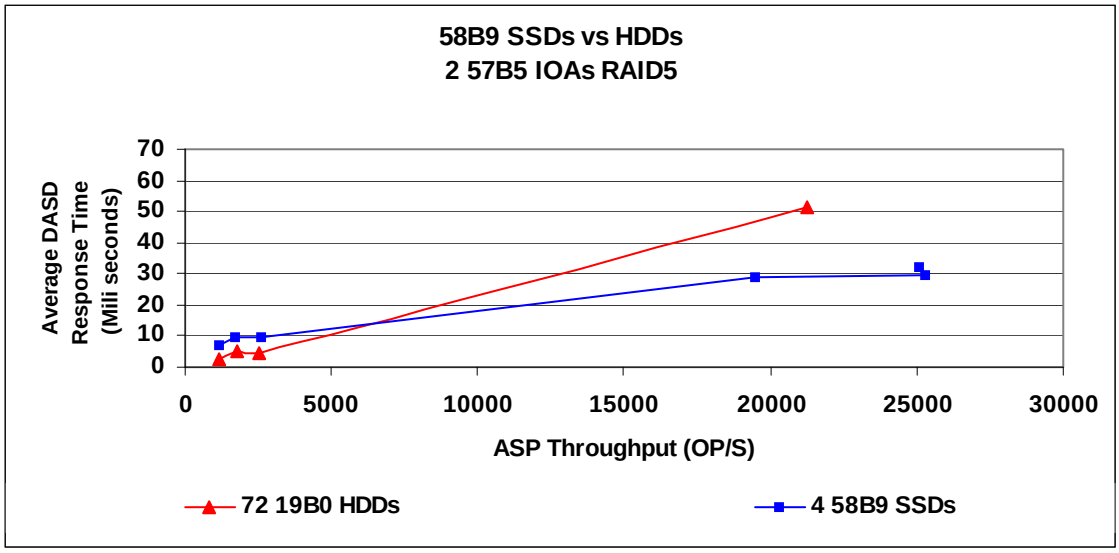
The following charts show the effect as the number of 58B9 devices was increased.





The new 589B SSDs have increased write capabilities as well as increased read capabilities. Previously, our recommendation was to place heavily read DB objects on SSDs to increase overall performance. This continues to be a valid approach for SSD usage. However with the improved write characteristics of 589B SSDs, you may want to consider placing objects that are simply heavily used by your application environment. This data shows that a single 58B9 can approximately match the speed of 20 15k RPM HDDs

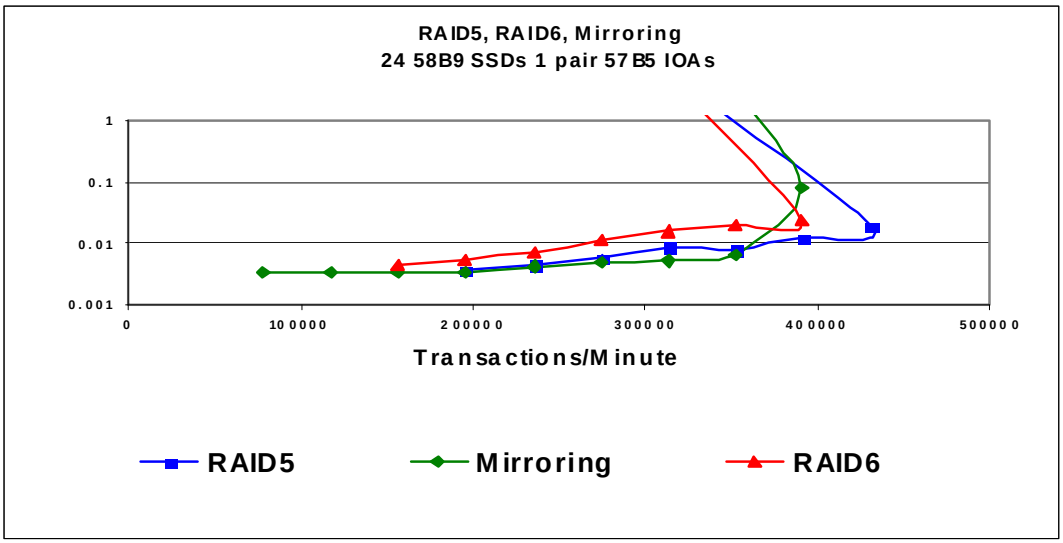


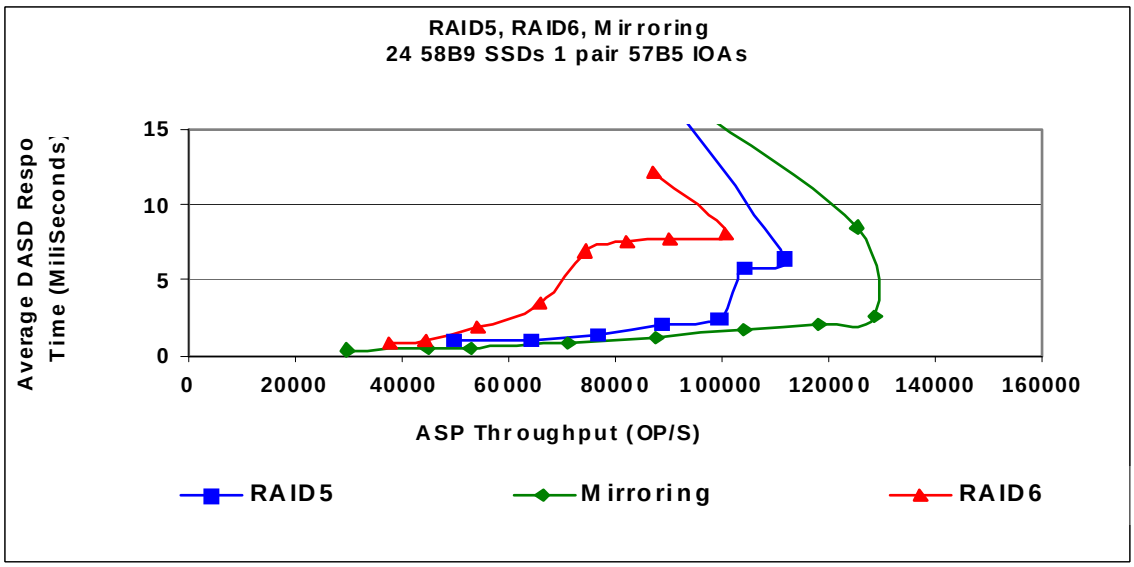


RAID6 has been enhanced for use with SSDs. The latest IOA PTF will be needed to open up the performance benefits.

When we use the maximum number of devices supported by the 57B5 IOA, the IOA itself can become the limiting factor for performance.

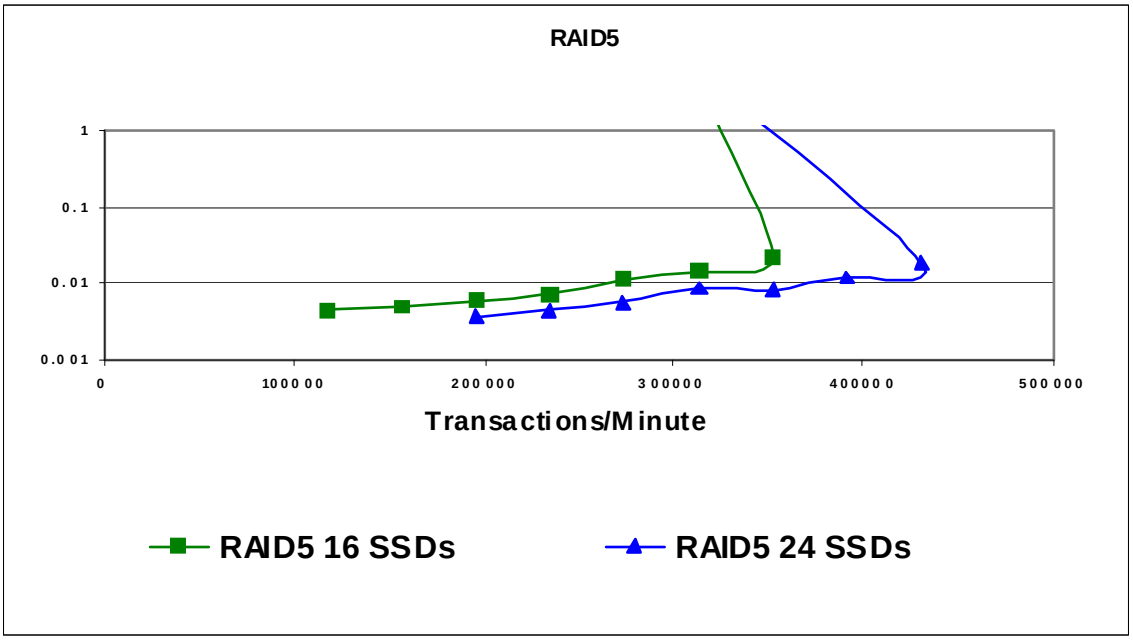
The charts below show the performance characteristics observed when we used the DASDIO workload (4 to 12k operations) on the different RAID types. For this workload the transactions per/min were similar for all of the RAID configurations.

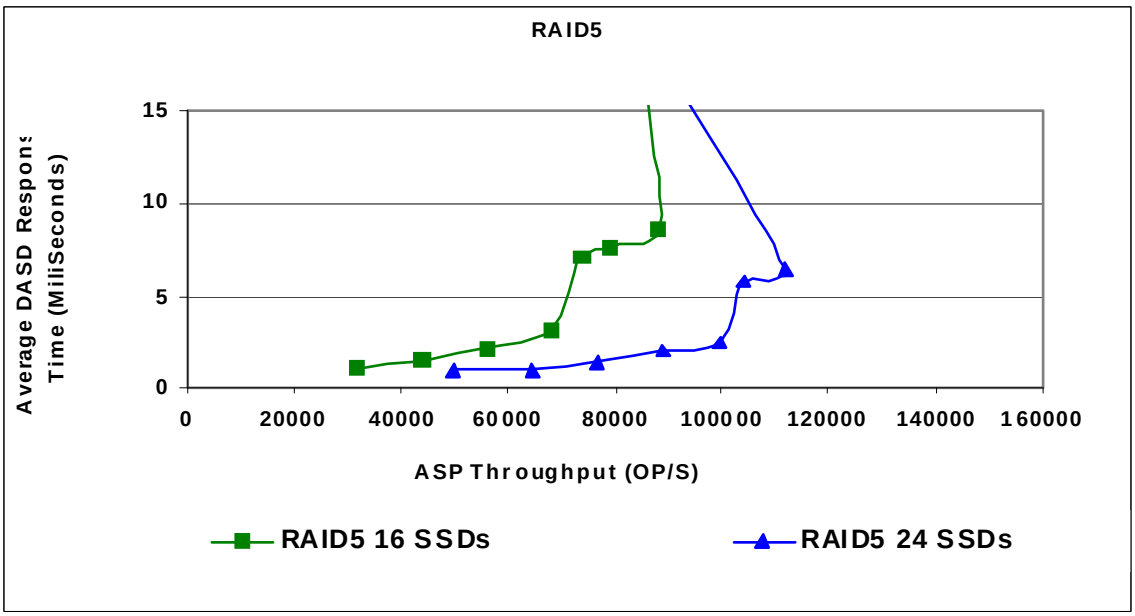




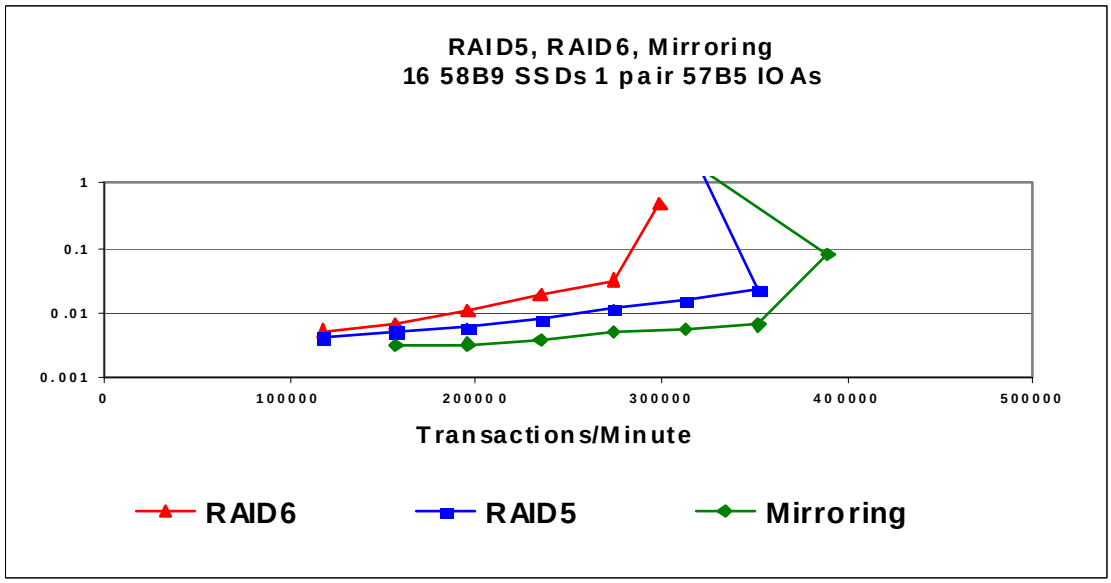
Next we chose to use 60 to 70% of the IOA, to help assure that the IOA would not become the performance limit but instead putting that on the SSD devices. This gives us a better view of the workload performance with the chosen RAID protection.

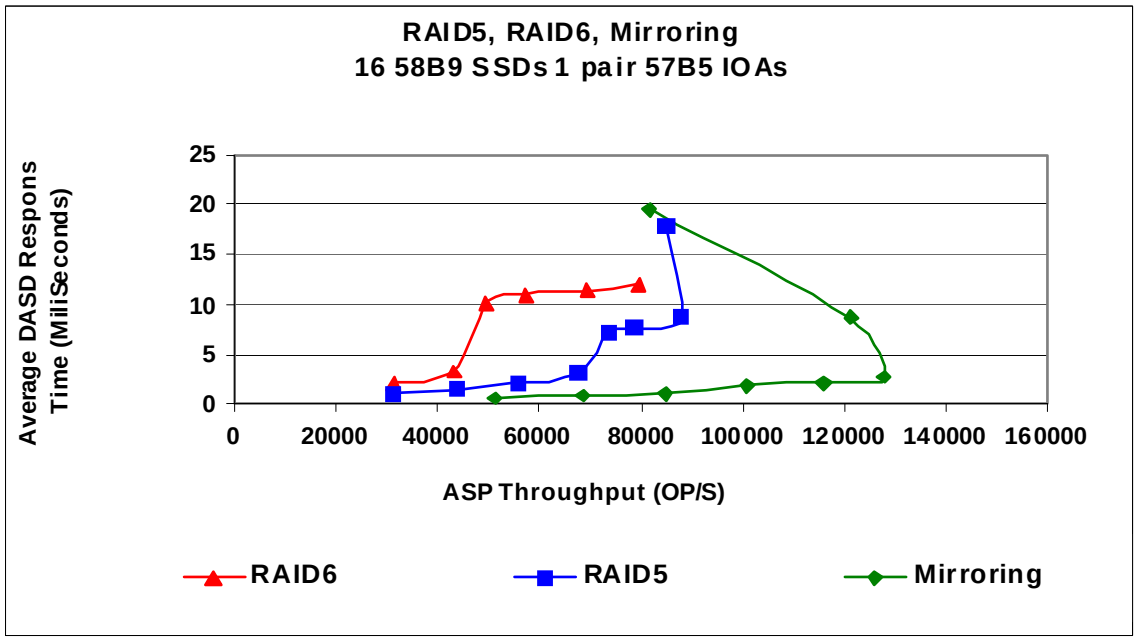
Choosing RAID5 as my base I changed from 24 to 16 SSDs. This place my IOA utilization more in line with the flow I desired.



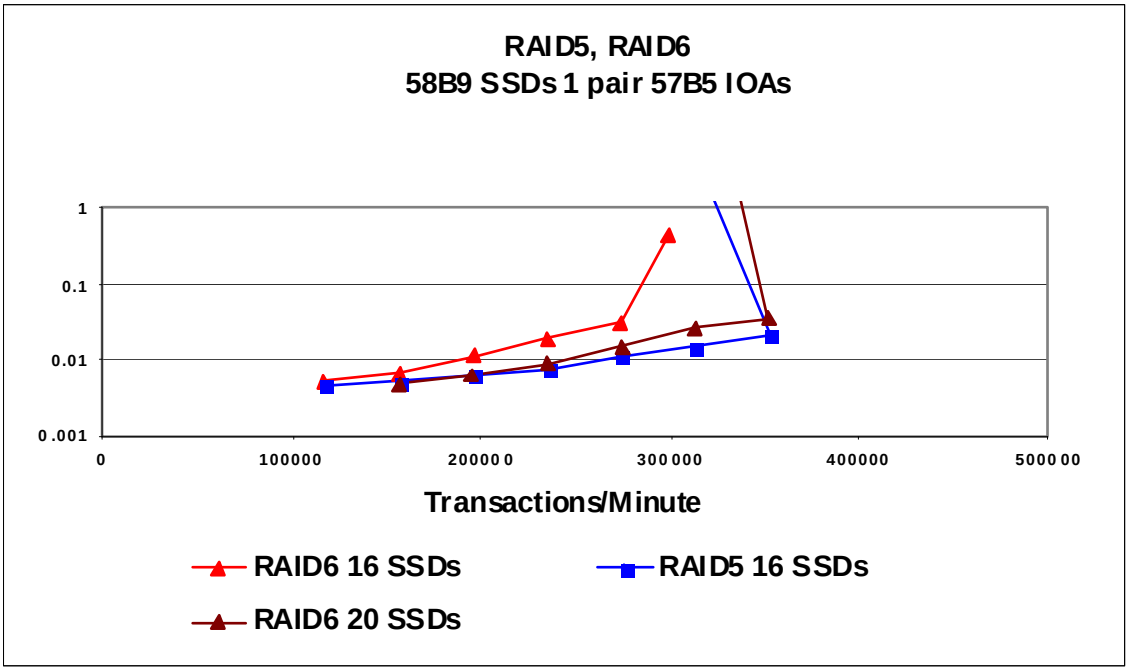


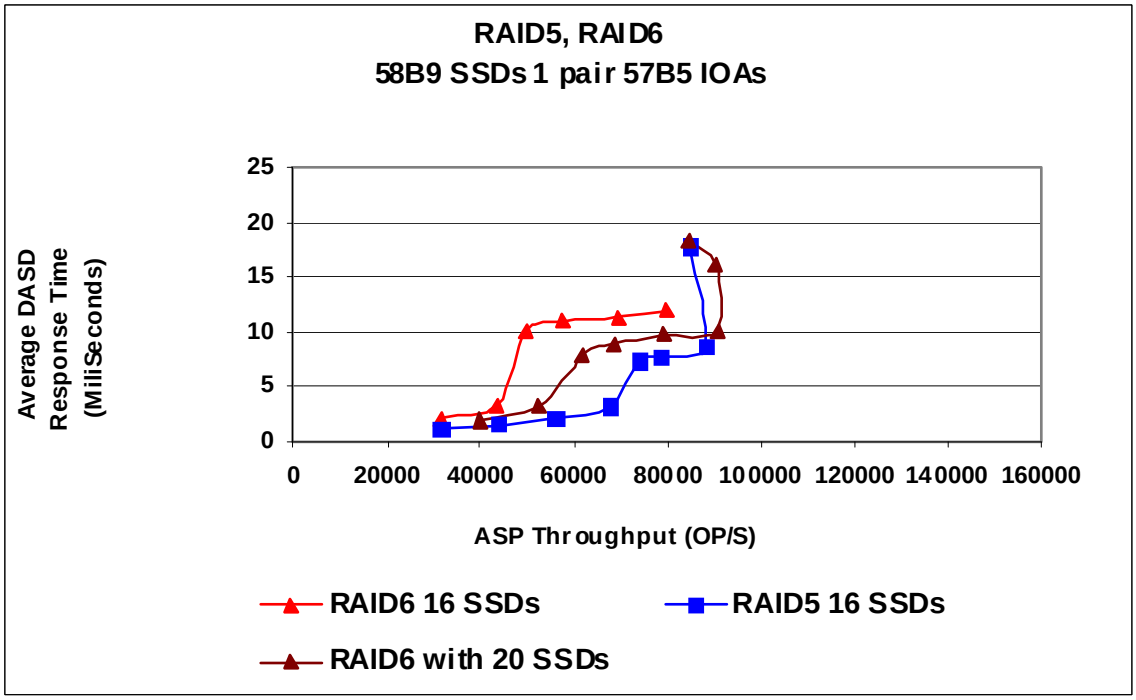
Using 16 SSD devices we were better able to observe the difference between the RAID5 RAID6 and IBMi Mirroring protection.





Then we increased the number of devices for RAID6 to achieve the utilization I desired. Giving me the number of devices to place on an IOA pair for this workload, to make the most of my IOA depending on the RAID protection I have chosen.





4.6.1 58B8/58B9 Solid State Devices on the 8205-E6C internal 57CB IOA

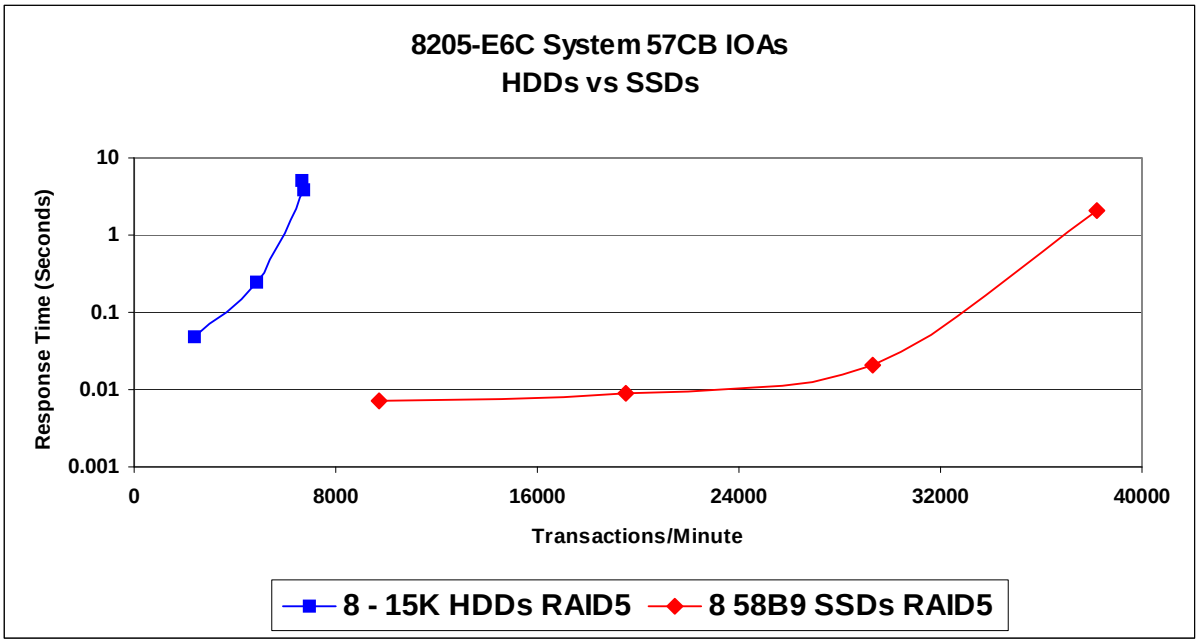
8502-E6C 57CB IOA comparing HDDs and SSDs

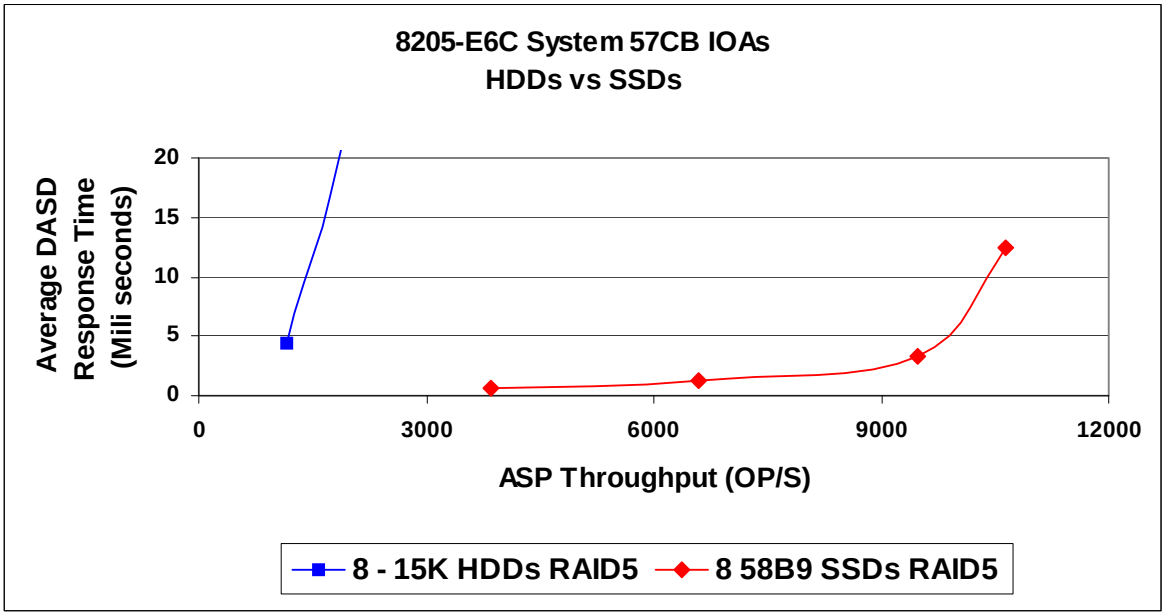
As customers compare their system needs the question has come up, “How do newer SSDs perform on the internal IOAs with smaller caches ?”

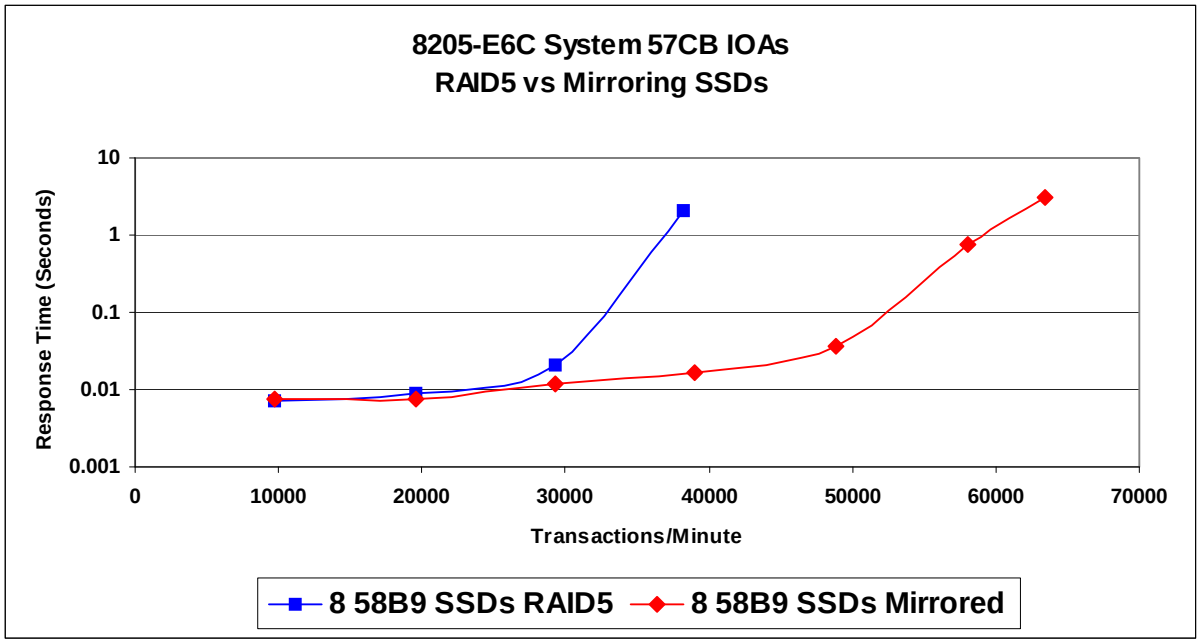
The following charts show comparisons using the DASDIO workload with 15K HDDs and the 58B9 SSDs. Even with the smaller cache of the IOA the performance of the 58B9 SSDs is clear, if you need performance with a small foot print, SSDs are a strong option to consider.

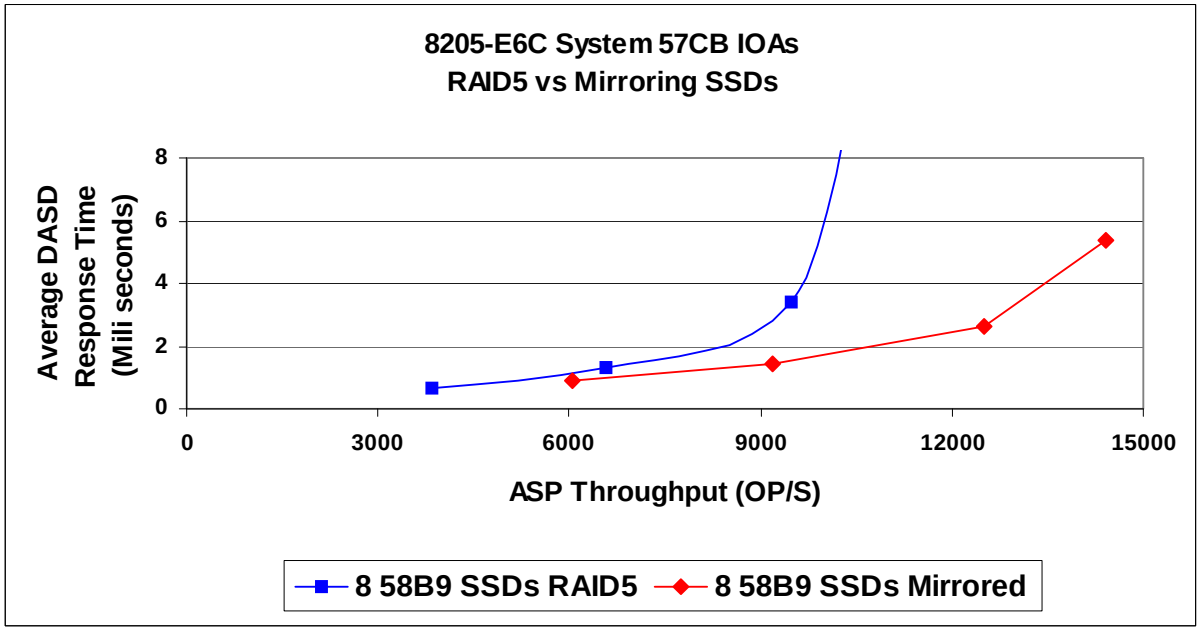
The first charts compare RAID5 15K HDDs to RAID5 58B9 SSDs. RAID5 allows the system to make the most out of the space available on the devices, but the cache size of the IOA can be a performance inhibitor. Even with the low cache size of the IOA the RAID5 58B9 SSDs are a good performance value.

The next set of charts show that if the space available on the devices is not the top issue, giving up more of the space to mirror the devices can provide better performance

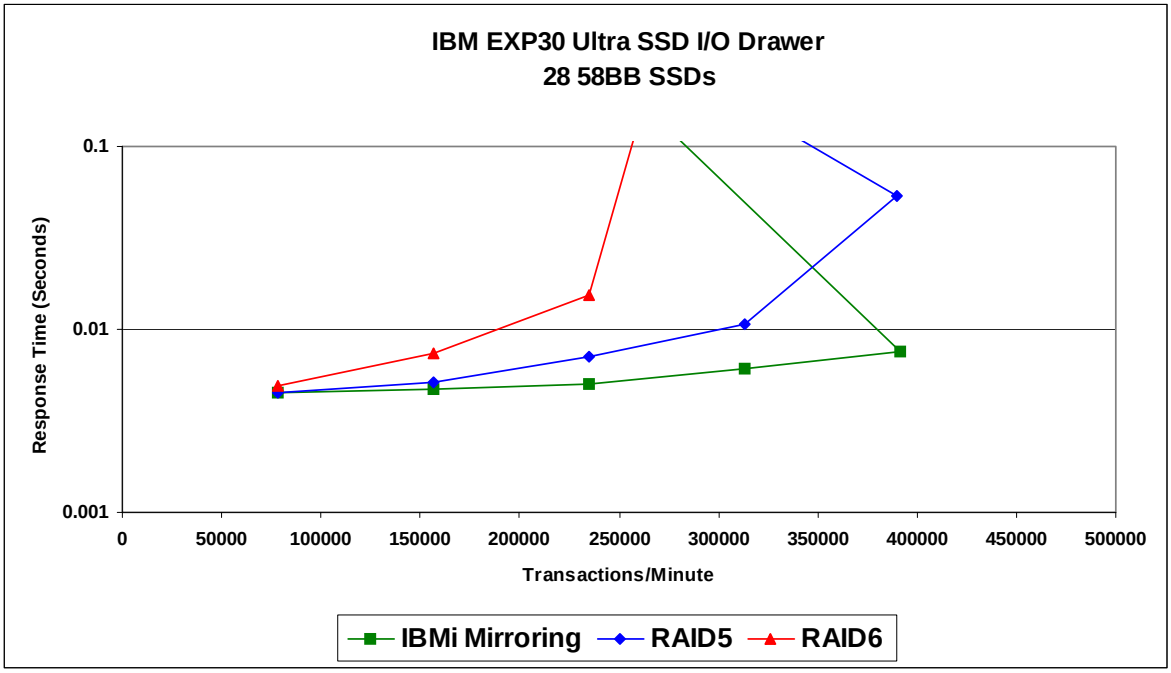


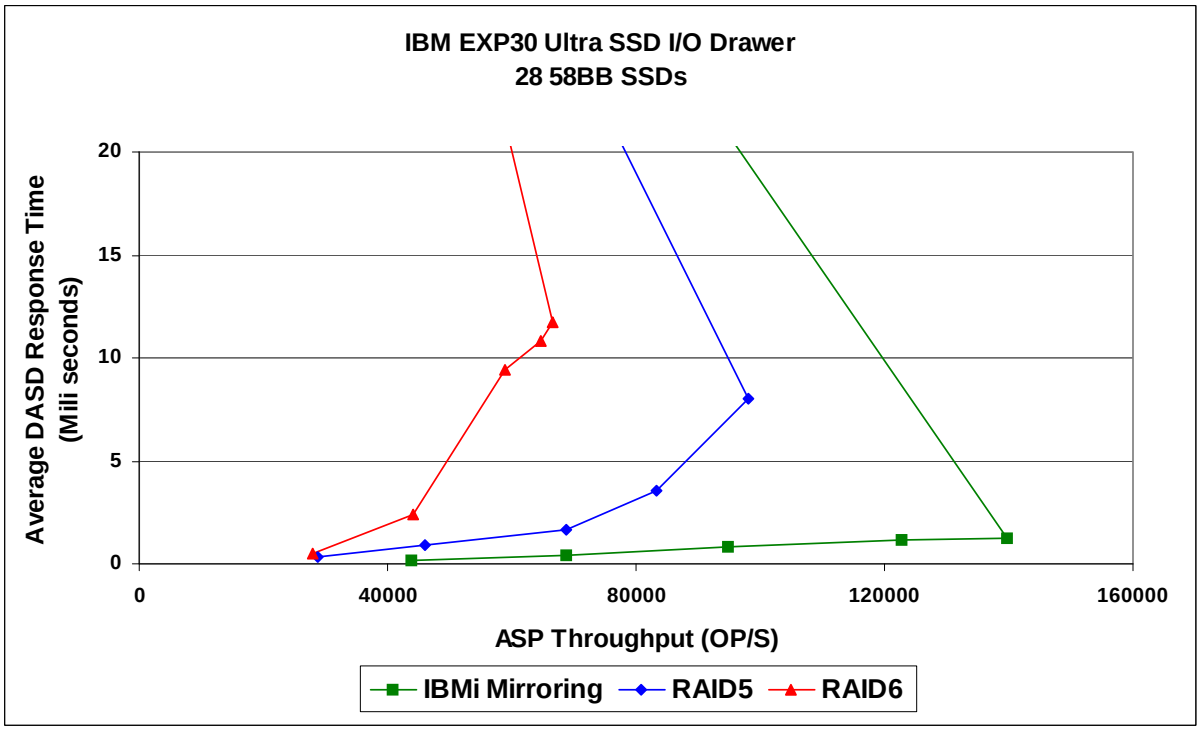


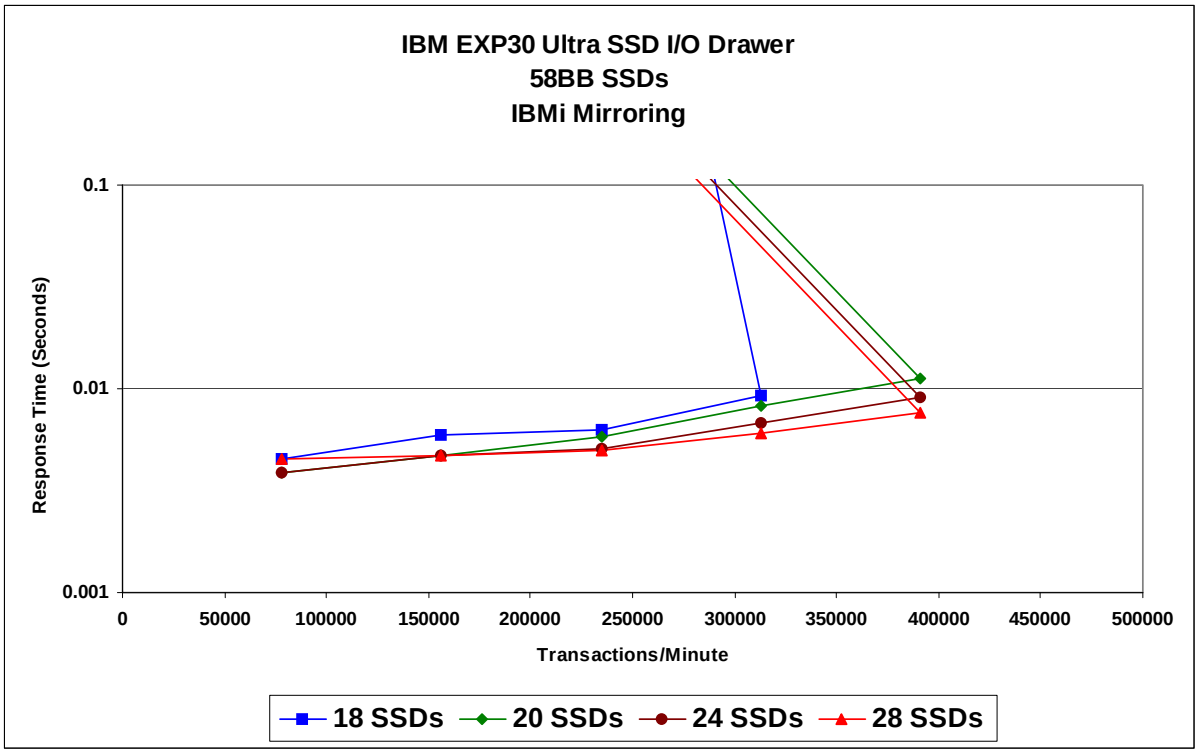


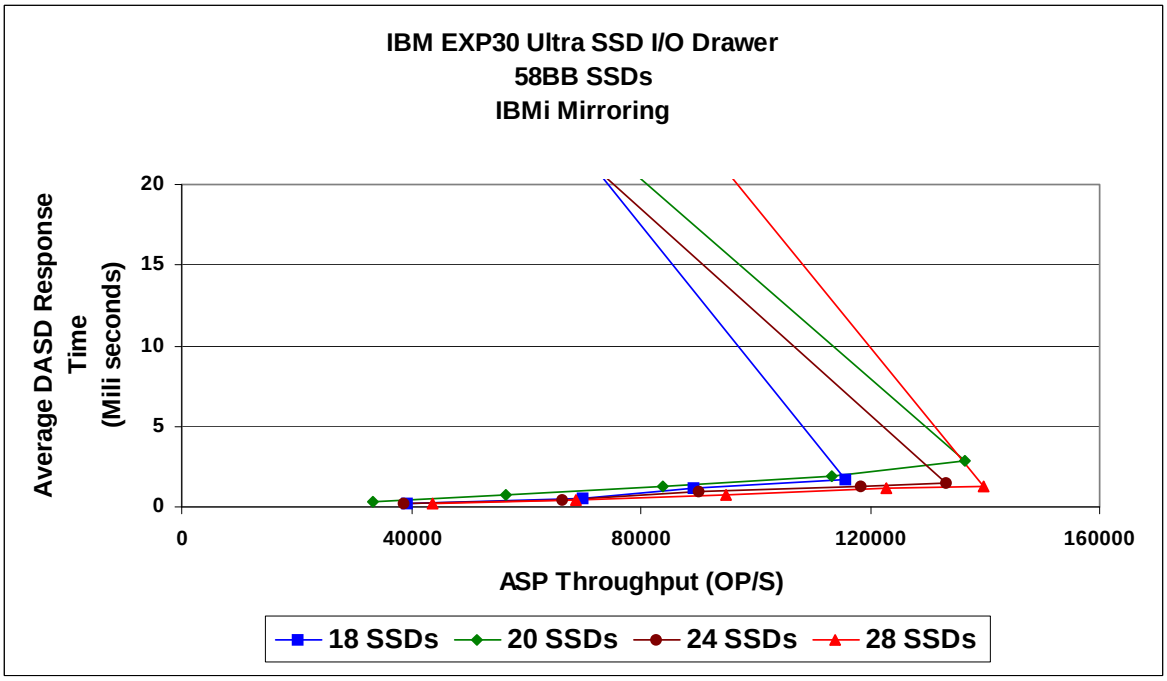


4.7 EDR1 - IBM EXP30 Ultra SSD I/O Drawer with 58BB SSDs









Chapter 5. SAN - Storage Area Network (External Storage) Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

5.1 DS5300 on IBM i

With more complex customer storage requirements, Storage Area Network (SAN) solutions are becoming the de facto standard. IBM i Performance would like to present the DS5300 Disk System performance measurements. The DS5300 is an excellent Mid to High range storage system that increases performance and capacity relative to the high end DS4000 series by leveraging the latest software and hardware technology. The DS5300 is perfectly suited to leverage the benefits inherent in the 8 Gbps Fiber Channel technology with sixteen 4 Gbps host channels to quickly transfer data, as well as, the 16 GB of physical cache memory (8 GB per controller). The DS5300 can access a total of 28 x EXP5000, EXP5060 or EXP810 expansion drawers for a total of 448 attached disk drives.

Beginning in IBM i 6.1 with the 6.1.1 LIC refresh , IBM i operating system will support DS5300 natively attached. A DS5300 storage solution will attach directly to an IBM i system. This will be an advantage for customers who do not currently run VIOS, and it could save system hardware resources since processors and memory resources will not be required for the VIOS partition. Performance running VIOS attached or natively attached DS5300 is similar.

5.1.1 DS5300 VIOS Attached

Refer to the following paper for performance information on DS5300 VIOS attached:
IBM® System Storage™ DS5300 - Performance Results in IBM i Power Systems Environment
http://www.ibm.com/systems/resources/ds5300performance.pdf

There are also some general VIOS concepts and recommendations to read over in the VIOS chapter which would apply to DS5300 VIOS attached storage.

5.1.2 DS5300 Native Attached

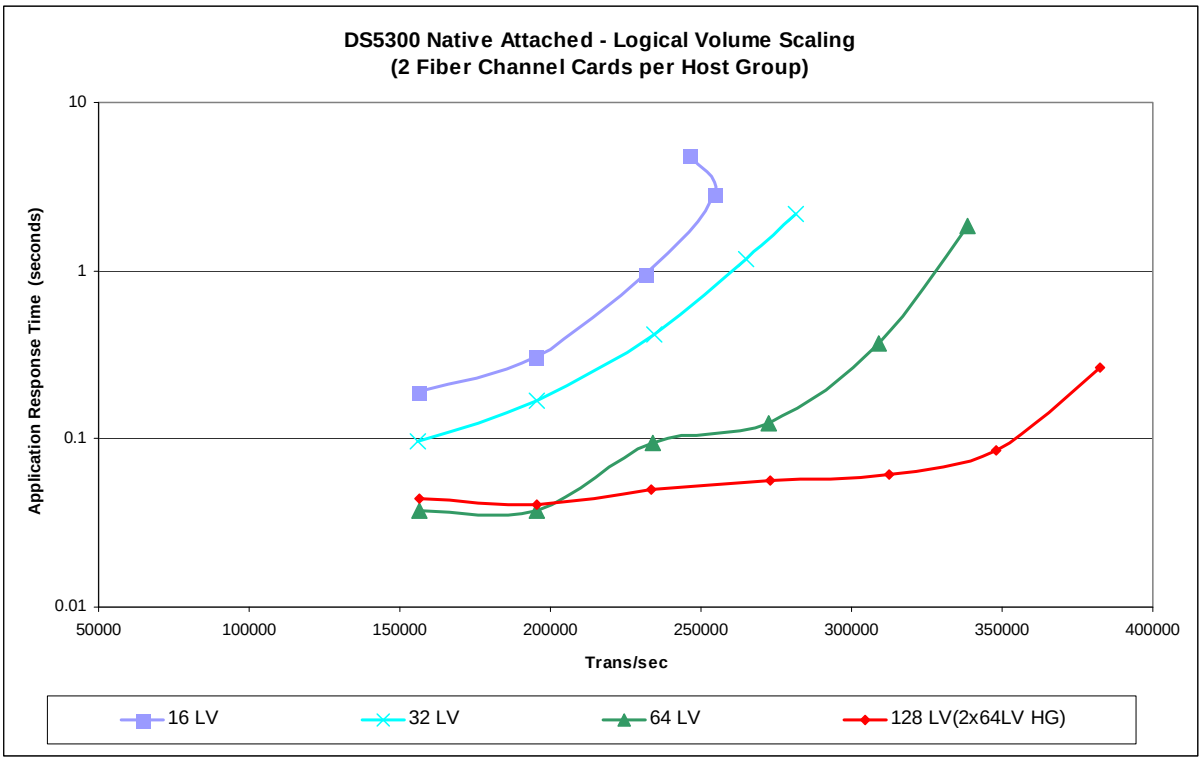
The following tests were run using a DS5300 with 16GB of cache, 8GB of mirrored memory per controller. The system had 8 expansion drawers (EXP5000) with a total of 128 DDMs mirrored (RAID 10). There were 16 RAID arrays created in the Storage manager which used 1 disk from each expansion drawer for a total of 8 physical disk in each array set. The logical volumes created over the 16 arrays were added to an ASP that contained the database under test. Typically, we had two fiber channel cards with 4 ports attached to each DS5300 host group. A DS5300 host group is a method to group disk on the DS5300 over specific physical fiber channel lines. The host group(s) are mapped to a partition by the slots being assigned to a particular partition.

The following charts show some of the performance characteristics of we observed running our Commercial Performance Workload(CPW) in our test environment. CPW is an internal tool used for Online Transaction Processing (OLTP) measurements. This workload is characterized by many jobs running brief database transactions in an environment that is dominated by IBM system code performing these database operations. Although the CPW workload is typically used to represent the relative performance of a complex of processors, it does a significant number of read and write DASD accesses

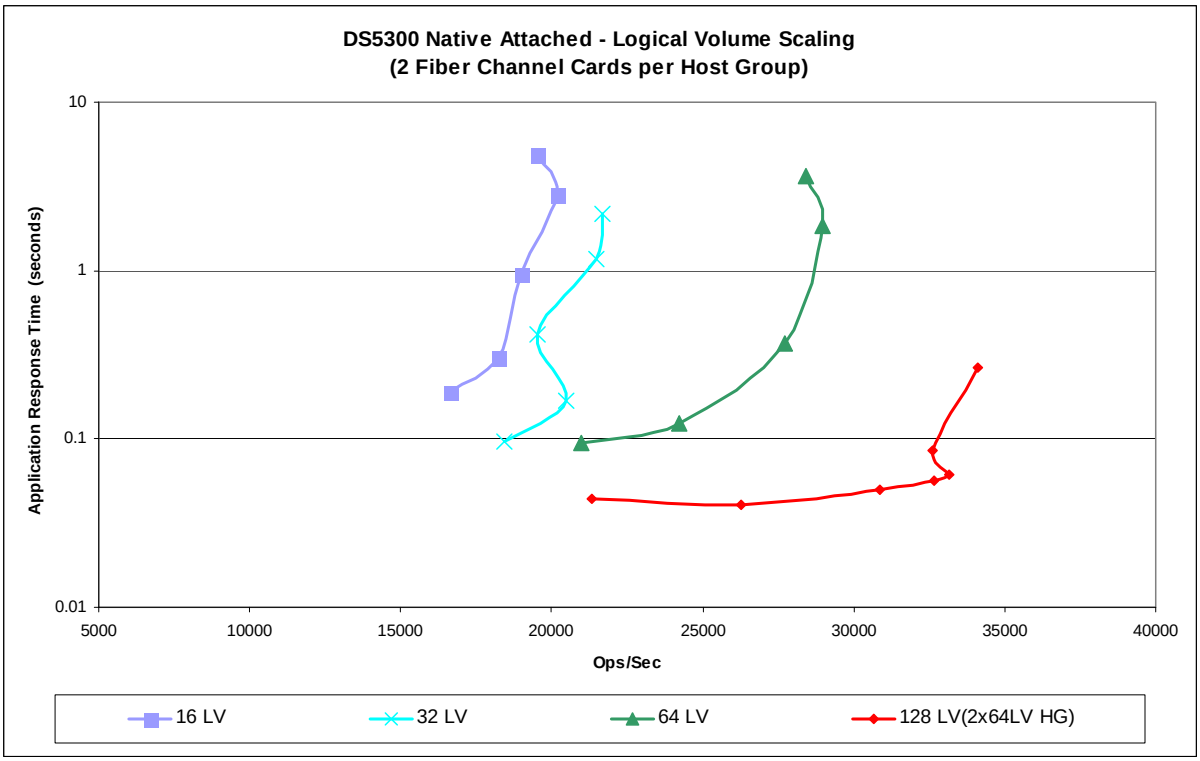
capable of saturating the I/O subsystem making CPW particularly useful in disk characterization. Your results may vary based on the characteristics of your workload. A description of the Commercial Performance Workload can be found in appendix A of the Performance Capabilities Reference.

5.1.3 DS5300 Native Attached Results*(Database only on DS5300)*

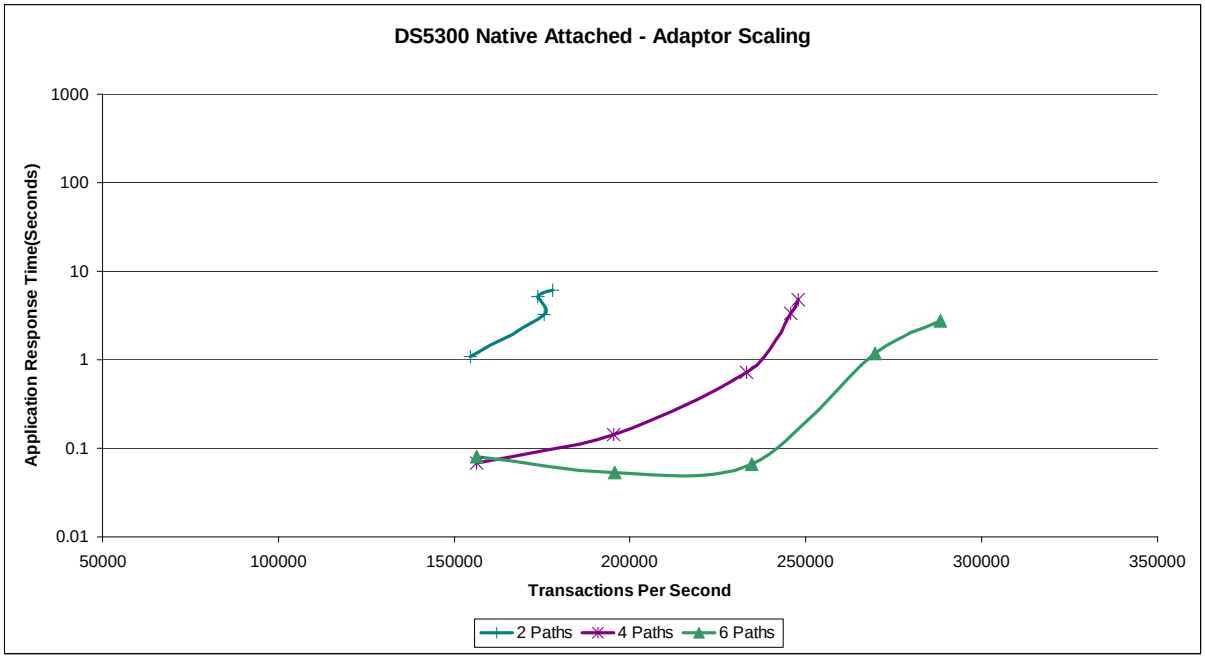
The first set of charts shows the effects of scaling logical volumes on natively attached IBM i. The maximum logical volumes allowed in a DS5300 host group is 64. The 128 logical volume test case below is achieved with 2x-64 logical volume host groups, and 4 fiber channel dual-port cards which enabled a total of 8 paths.



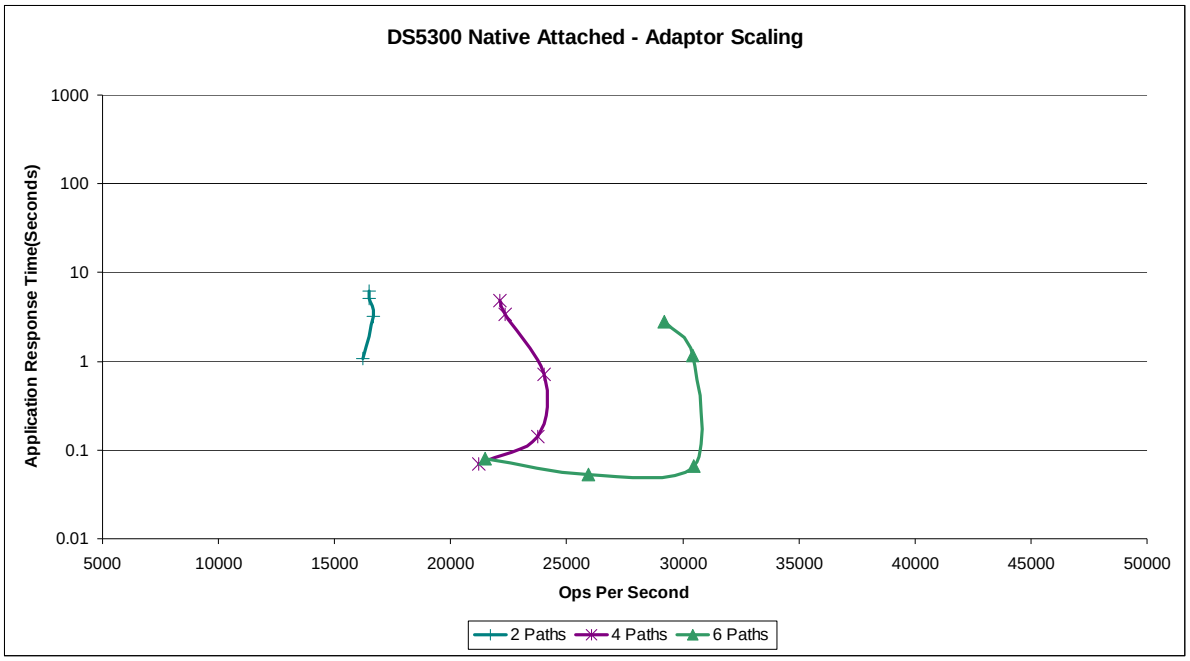
Here is a second chart which shows logical volume scaling but comparing ops/sec.



Paths are fiber channel connections between the DS5300 and the IBM i Power system. In the previous example there was 2 dual port fiber channel cards, or a total of 4 paths. The example below shows the performance of scaling paths while keeping the number of logical volumes constant at 64.



Here is a second chart which shows path scaling but comparing ops/sec.



5.2 External Storage Best Practices

- **LUNS -**
  IBM i performs the best when a large number of LUNS are used. In order to achieve the best performance use 64 Logical Volumes in a DS5300 host group. (The maximum number of LUNS supported per Fibre Channel port in a IBM i Partition is 64)
- **Fiber Channel Connections -**
  IBM i native attached performs the best when a larger number of fiber channel paths are attached. In order to achieve the best performance use at least 6 fiber channel paths in a DS5300 host group.
- **DS5300 Host Groups -**
  64 logical volume should be a minimum for optimal performance, Each LUN should have a minimum of 4 paths.
- **Active/Passive Paths are equally distributed -**
  The paths should be equally distributed with an equal number of active/passive paths for each disk unit. This can be seen in STRSST>Work with Disk Units->Display Disk Configuration->Display Disk Path Status

2	19	Y63C4AF297DC	D818	099	DMP008	Passive
					DMP007	Active
2	20	Y2974AF29788	D818	099	DMP010	Passive
					DMP012	Active

- **DS5300 Controller A & B balanced -**
  In order to balance the arrays on the DS5300 controllers. Place the first half of the arrays on Controller A. The second half of the arrays on Controller B. In the test done in the lab we had 16 arrays. Arrays 1-8 were placed on controller A. Arrays 9-16 were placed on controller B.
- **Multipath -**
  Multipath requires a minimum of 2 physical adapters. Multipath on a single adapter is not supported.

5.3 External Resources

There are many factors to consider when looking at external storage options, you can get more information through your IBM representatives and the white papers that are available at the following locations.

IBM® System Storage™ DS5300 - Performance Results in IBM i™ Power Systems Environment (VIOS attached DS5300)
http://www.ibm.com/systems/resources/ds5300performance.pdf

IBM System Storage Product Guide
http://www.ibm.com/systems/resources/systems_storage_resource_pguide_prodguidedisk.pdf

IBM System Storage DS5000 series
http://www.ibm.com/systems/storage/disk/ds5000/index.html

IBM i and Midrange External Storage
http://www.redbooks.ibm.com/abstracts/sg247668.html?Open

IBM XIV® Storage System (VIOS attached)
Contact XIV Technical Sales

PowerVM Migration from Physical to Virtual Storage
http://www.redbooks.ibm.com/redpieces/abstracts/sg247825.html?Open

Chapter 6. VIOS

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

6.1 VIOS and IVM Considerations

Beginning in iV6R1M0, IBM i operating system will participate in a new virtualization strategy by becoming a client of the VIOS product. Customers will view the VIOS product two different ways:

- On blade products, through the regular configuration tool IVM (which includes an easy to use interface to VIOS).
- On traditional (non-blade) products, through a combination of HMC and the VIOS command line.

The blade products have a simpler interface which, on our testing, appear to be sufficient for the environments involved. On blades, using the IVM interface, customers are restricted to a single VSCSI and 16 logical units (which IBM i operating system perceives as if they were physical drives). This substantially reduces the number and value of tuning options.
NOTE: It is possible for blade-based customers to use the VIOS command line interface to create more VSCSI interfaces and map 16 LUNs to each VSCSI created. See VIOS configuration manuals for information on creating VSCSI and maping LUNs from the command line.

Customers should strongly consider their disk requirements here and consult with their support teams before ordering. Customers with more sophisticated disk-based requirements (or, simply, larger numbers of disks) should choose systems that allow a greater number of LUNs and thereby enable more substantial tuning options provided from the VIOS command line. No hard and fast rules are possible here and we again emphasize that one consult with their support team on what will work for them.

6.2 General VIOS Considerations

6.2.1 Generic Concepts

***520 versus 512*.** Long time IBM i operating system users know that IBM i operating system disks are traditionally configured with 520 byte sectors. The extra eight bytes beyond the 512 used for data are used for various purposes by Single Level Store.

For a variety of reasons, VIOS will always surface 512 byte sectors to IBM i operating system whatever the actual sector size of the disk may be. This means that 520 byte sectors must be emulated within 512 byte sectors when using disks supported by VIOS. This is done, simply enough, by reading nine 512 byte data sectors for every eight sectors of actual data and placing the Single Level Store information within the extra sector. Since all disk operations are controlled by Single Level store in an IBM i operating system there are no added security implications from this extra sector, provided standard, sensible configuration practices are followed just as they would be for regular 520 byte devices.

However, reading nine sectors when only eight contain data will cost some performance, most of it being the sheer cost of the extra byte transfer of the extra sector. The gains are the standard ones of

virtualization -- one might be able to share or re-purpose existing hardware for System i's use in various ways.

Note carefully that some "512" byte sectored devices actually have a range of sizes like 522, 524, and others. Confusingly for us, the industry has gone away from strictly 512 byte sectors for some devices. They, too, have headers that consume extra bytes. However, as noted above, these extra bytes are not available for IBM i operating system and so, for our purposes, they should be considered as if they were 512 byte sectored, because that is what IBM i operating system will see. Some configuration tools, however, will discuss "522 byte" or whatever the actual size of the sectors is in various interfaces (IVM users will not see any of this).

VIOS will virtualize the devices. Many configuration options are available for mapping physical devices, as seen by VIOS, to virtual devices that VIOS will export to DST and Single Level Store. Much more of this will be done by the customer than was done with internal disks. Regardless of whether the environment is blades or traditional, it is important to make good choices here. Even though there is much functional freedom, many choices are not optimized for performance or optimized in an IBM i operating system context. Moreover, nearly as a matter of sheer physics, some choices, once made, cannot be much improved without very drastic steps (e.g. dedicating the system, moving masses of data around, etc.). Choosing the right configuration in the first place, in other words, is very important. Most devices, especially SAN devices, will have “Best Practices” manuals that should be consulted.

6.2.2 Generic Configuration Concepts

There are several important principles to keep track of in terms of getting good performance. Most of the following are issues when the disks are configured. A great many problems can be eliminated (or, created) when the drives are originally configured. The exact nature of some of these difficulties might not be easily predicted. But, much of what follows will simply avoid trouble at no other cost.

1. ***Ensure that RAID protection is performed as close to the physical device as possible .*** This is typically done out at an I/O adapter level or on the external disk array product. This means that either the external disk's configuration tools or (for internal disks assigned to VIOS) VIOS' tools will be used to create RAID configurations (RAID5, RAID10, or RAID1). When this is done, as far as IBM i operating system disk status displays are concerned, the resulting virtual drives appear to be "unprotected." It might be superficially reassuring to have IBM i operating system do the protection (if IBM i operating system even permits it). WRKDSKSTS would then show the protection on that path. DST/SST disk configuration functions would show the protection, too. However, it is better to put up with what appears to IBM i operating system's disk status routines to be unprotected devices (which are, after all, actually protected) than to take on the performance problems of doing this under IBM i operating system. RAID recovery procedures will have to be pursued outside of IBM i operating system in any event, so the protection may as well go where the true physicality is understood (either in VIOS or the external disk array product).
Note also that you also want to configure things so that the outboard devices, rather than VIOS, do the RAID protection whenever possible. This enables I/O to flow directly from the device to IBM i operating system as directed by VIOS.

High Availability scenarios also need to be considered. In some cases, to enable appropriate redundancy, it may be necessary to do the protection a little farther away from the device (e.g. spread over a couple of adapters) so as to enable the proper duplexing for high availability. If this applies to you, consult the

documentation. Some external storage devices have extensive duplexing within themselves, for instance, which could allow one to keep the protection close to the device after all.

2. ***Recognize that Internal Disks remain the "gold standard" for performance.*** We have consistently measured external disks as having less performance than 520 byte, internally attached disks. However, the loss of throughput, with proper configuration, is not a major concern. What is harder to control is response time. If you have sensitivity to response time, consider internal disks more strongly.

3. ***Prefer external disks attached directly to IBM i operating system over those attached via VIOS*** This is basically a statement of the fiber channel adapter and who owns it. In some cases, it affects which adapter is purchased. If you do not need to share a given external disk's resources with non-IBM i operating system partitions, and the support is available, avoiding VIOS altogether will give better performance. First, the disks will usually have 520 byte support. Second, the IBM i operating system support will know the device it is dealing with. Third, VIOS will typically run as a separate partition. If you run VIOS as your first shared partition, simply turning on shared support costs about five to eight percent overall. The alternative, a dedicated partition for VIOS, would be a nice thing to avoid if possible. If you would not have used shared processor support otherwise, or would have to give VIOS a whole processor or more otherwise, this is a consideration.

4. ***Prefer standard IBM i operating system internal disks to VIOS internal disks***. This describes who should own a given set of internal disks. If there is a choice, giving the available internal disks to IBM i operating system instead of going through VIOS will result in noticeably better performance. VIOS is a better fit for external disk products that do not support the IBM i operating system 520 byte sector. The VIOS case would include internal disks that came originally from pSeries or System p. However, one should investigate those devices also. If those devices support 520 byte sectors (or, alternatively, if it is stated they are supported by IBM i operating system), they should be reconfigured instead as native IBM i operating system internal disks. It should be exceptional to use VIOS for internal disks.

5. ***Prefer RAID 1 or RAID 10 to RAID 5***. We are now beginning to generally recommend RAID 1 ("mirroring") or RAID 10 (a "mirroring" variant) for disks generally in On-line Transaction Processing (OLTP) environments. OLTP environments have long had to deal with configurations based on total arm count, not capacity as such. If that applies to you, you have extra space that is of marginal value. Those in this situation can nowadays use the same number of arms deployed as RAID 1 or RAID 10 to gain increased performance. This is at least as true for external disks as it is for internal disks. Note that in this recommendation, one deploys the same arm count -- just deploys them differently, trading unused space for performance. Also note that if one goes this route, two physical disks per RAID 10 or RAID 1 set is better than a larger number of disks per RAID 1 or RAID 10 set. (See also “Ensure, within reason” below).

6. ***For VIOS, Prefer External Disks (SAN disks) to Internal Disks***. SAN disks will have greater flexibility and better tuning options than internal disks. Accordingly, when there is a choice, VIOS is best used for external disks.

7. ***Separate Journal ASPs from other ASPs.*** Generally, we have long recommended that a given set of data base files (aka SQL tables) keep its set of journal receivers in a separate ASP from the data base ASP or ASPs. With VIOS, we recommend that this continue to the extent feasible. It may be necessary to share things like Fibre Channel links, but it should be possible to have separate physical devices at the very least. To the extent possible, arrange for journal to use its own internal buses also (of whatever sort the device provides).

8. ***Ensure, within reason, a reasonable number of virtual disks are created and made available to IBM i operating system.*** One is tempted to simply lump all the storage one has in a virtual environment into a couple (or even one) large virtual disk. Avoid this if at all possible.
There is a great deal of variability here, so generalizations are difficult. However, in the end, favor virtual disks that are within a binary order of magnitude or two of the physical disk sizes. Make each them as close to the same size if possible. In any case, strive to have half a dozen or more in an ASP if you can. Years of system tuning (at all levels) tacitly expect a reasonable number of devices, so it makes sense to provide a bunch. You don't need a count larger than the physical device count, however, unless the device count is very small.

9. ***Prefer Symmetrical Configurations***. To the extent possible, we have found that physical symmetry pays off more than we have seen before. Balancing the number of physical disks as much as possible seems to help. Strive to have uniform LUN sizes, uniform number of disks in each RAID set, balance (at least at the static configuration level) between the various internal and external buses, etc. To the extent practical, the user should strive for even numbers of items.

10. ***In general, do not share the same physical disk with multiple partitions.*** Only If you are running some minimal IBM i operating system partition (say, a very small Domino partition or perhaps a middle tier partition that has no local data base), should you consider strategies where IBM i operating system is sharing physical disks with other partitions. For more traditional application sets (whether a traditional system or a blade) you'll have a data base or large enough data contents generally to give each IBM i operating system partition its own physical devices. Once you get to multiple devices, sharing them with other partitions will lead to performance problems as the two partitions fight (in mutual ignorance) for the same arm, which may increase seek time (at least) a little to a lot. Service time could be adversely affected as well.

11. ***To the extent possible, think multiple VIOS partitions for multiple IBM i operating system partitions.*** If the physical disks deserve segmentation, multiple VIOS partitions may also be justified. The main issue is load. If the IBM i operating system partitions are small (under two CPUs), then you're probably better off with a shared VIOS partition hosting a couple of small IBM i operating system partitions. As the IBM i operating system partitions grow, it will be possible to justify dedicated VIOS partitions. Our current measurements suggest one VIOS processor for every three IBM i operating system processors, but this will vary by the application.

6.2.3 Specific VIOS Configuration Recommendations -- Traditional (non-blade) Machines

1. ***Avoid volume groups if possible***. VIOS "hdisks" must have a volume identifier (PVID). Creating a volume group is an easy way to assign one and some literature will lead you to do it that way. However, the volume group itself adds overhead for no particular value in a typical IBM i operating system context where physical volumes (or, at least, RAID sets) are exported as a whole without any sort of partitioning or sub-setting. Volume groups help multiple clients share the same physical disks. In an IBM i operating system setting, this is seldom relevant and the overhead volume groups employ is therefore not needed. It is better to assign a PVID by simply changing the attribute of each individual hdisk. For instance, the VIOS command: `chdev -dev hdisk03 -attr pv=yes` will assign a PVID to hdisk3.

2. For VIOS disks, ***use available location information to aid your RAID planning***. To obtain RAID sets in IBM i operating system, you simply point DST at particular groups you want and IBM i operating system decides which disks go together. Under VIOS, for internal disks, you have to do this yourself. The names help show you what to do. For instance, suppose VIOS shows the following for a set of internal disks:

```
Name      Location      State       Description       Size
pdisk0   07-08-00-2,0  Active      Array Member      35.1GB
pdisk1   07-08-00-3,0  Active      Array Member      35.1GB
pdisk2   07-08-00-4,0  Active      Array Member      35.1GB
pdisk3   07-08-00-5,0  Active      Array Member      35.1GB
pdisk4   07-08-00-6,0  Active      Array Member      35.1GB
pdisk5   07-08-01-0,0  Active      Array Member      35.1GB
pdisk6   07-08-01-1,0  Active      Array Member      35.1GB
```

Here, it turns out that these particular physical disks are on two internal SCSI buses (00 and 01) and have device IDs of 2, 3, 4, 5, and 6 on SCSI bus 00 and device IDs of 0 and 1 on SCSI bus 01. If this was all there was, a three disk RAID set of pdisk0, pdisk1, and pdisk5 would be a good choice. Why? Because pdisk0 and 1 are on internal SCSI bus 00 and the other one is one SCSI bus 01. That provides a good balance for the available drives. This could also be repeated for pdisk2, pdisk3, pdisk4, and pdisk6. This would result in two virtual drives being created to represent the seven physical drives. The fact that these are two RAID5 disk sets (of three and four physical disks, respectively) would be unknown to IBM i operating system, but managed instead by VIOS. One or may be two virtual SCSI buses would be required to present them to IBM i operating system by VIOS. A large configuration could provide for RAID5 balance over even more SCSI buses (real and virtual).

On external storage, the discussion is slightly more complicated, because these products tend to package data into LUNs that already involve multiple physical drives. Your RAID set work would have to use whatever the external disk storage product gives you to work with in terms of naming conventions and what degree of control you have available to reflect favorable physical boundaries. Still, the principles are the same.

3. ***Limited number of virtual devices per virtual SCSI adapter.*** You will have to configure some number of virtual SCSI adapters so that VIOS can provide a path for IBM i operating system to talk to VIOS as if these were really physical SCSI devices. These adapters, in turn, implement some existing rules, so that only 16 virtual disks can be made part of a given virtual adapter. You probably would not want to exceed this limit anyway. Note that the virtual adapters need not relate to physical boundaries of the various underlying devices. The main issue is to balance the load. You may be able to segregate data base and journal data at this level. Note also that in a proper configuration, the virtual SCSI adapters will carry command traffic only. The actual data DMA will be direct to the IBM i operating system partition.

4. ***VIOS and Shared Processors*****.** On the whole, dedicated VIOS processors will work better than shared processors, especially as the IBM i operating system partition needs three or more CPUs itself. If you do not need shared processors for other reasons, experiment and see if dedicated VIOS processors work better. In fact, it might be an experiment worth running even if you have shared processors configured generally.

5. ***VIOS and memory*****.** VIOS arranges for the DMA to go directly to the IBM i operating system memory (with the help of PHYP and IBM i operating system to ensure integrity). This means that actual data transfer will not go through VIOS. It only needs enough main storage to deal with managing disk traffic, not the data the traffic itself consumes. Our current measurements suggests that 1 GB of main storage is the minimum recommended. Other work suggest that unless substantial virtual LAN is involved, between 1 GB and 2 GB tends to suffice at the 1 to 3 CPU ratio we typically measured.

6. ***VIOS and Queue Depth***. Queue depth is a value you can change, so one can experiment to find the best value, at least on a per IPL basis. VIOS tends the set the queue depth parameter to smaller values. Especially if you follow our recommendations for the number of virtual disks, you will find values like 32 to work well for the device as a starting point. If you do that, you will also want to set the queue depth for the adapter (usually called num_cmd_elems) to its larger value, often 512. Consult the documentation.

6.3 IBM i operating system 5.4 Virtual SCSI Performance

The primary goal of virtualization is to lower the total cost of ownership of equipment by improving utilization of the overall system resources and reducing the labor requirements to operate and manage many servers.

With virtualization, the IBM Power Systems can now be used similar to the way mainframes have been used for decades, sharing the hardware between many programs, services, applications, or users. Of course, for each of these individual users of the hardware, sharing resources may result in lower performance than having dedicated hardware, but the overall cost is usually far less than when dedicating hardware to each user. The decision of using virtualization is therefore a trade-off between cost and performance.

IBM i operating system Virtual SCSI is based on a client/server relationship. A IBM i operating system Server partition owns the physical resources, and client partitions access the virtual SCSI resources provided by the IBM i operating system Server partition. The IBM i operating system Server partition has physically attached I/O devices and exports one or more of these devices to other partitions. The client partition is a partition that has a virtual disk and relies on the IBM i operating system Server partition to provide access to one or more physical devices. POWER5 and future POWER technologies provide

virtual SCSI support for AIX 5L V5.3 and Linux. Previous POWER technology supported Linux virtual SCSI.

The performance considerations that we detail in this section must be balanced against the savings made on the overall system cost. For example, the smallest physical disk that is available to the IBM i operating system is 70 GB. An AIX or Linux operating system requires only 4 GB of disk. If one disk is dedicated to the operating system, nearly 95% of this physical disk space is unused. Furthermore, the system disk I/O rate is often very low. With the help of IBM i operating system Virtual SCSI, it is possible to split the same disk into 9 virtual disks of about 8 GB each. If each of these disks is used for installation of the operating system, you can support nine separate instances of the operating system, with nine times fewer disks and perhaps as many physical SCSI adapters. Compare these savings with the extra cost of processing power needed to handle the virtual disks.

Enabling IBM i operating system Virtual SCSI results in using extra processing power compared to directly attached disks, due to extra POWER VIO activity. Depending on the configuration, this may or may not yield the same performance when comparing virtual hosted disk devices to physically attached SCSI devices. If a partition has high performance and disk I/O requirements that justify the cost of dedicated hardware, then using virtual SCSI is not recommended. However, partitions with non-critical performance and low disk I/O requirements often can be configured to use virtual SCSI, which in turn lowers hardware and operating costs.

In the test results that follow, we see the CPU required for IBM i operating system Virtual SCSI server and the benefits of the IBM i operating system Virtual SCSI implementation should be assessed for a given environment. Simultaneous multithreading should be enabled in a virtual hosted disk environment. For most efficient virtual hosted disk implementation with larger IO loads, it may be advantageous to keep the IBM i operating system Virtual SCSI Server partition as a dedicated processor. Processor micro partitioning should be used with low IO loads or with workloads which are not latency dependent.

Virtual storage can be created in an ASP using the CRTNWSSTG and linked using the CRTNWSD commands. The disk can be manipulated in the client AIX or Linux partition the same as an ordinary physical disk. Some performance considerations from dedicated storage are still applicable when using virtual storage, such as spreading ASP’s across multiple disks on multiple RAID adapters so that parallel access is possible. From the server's point of view, a virtual drive can be served using an entire ASP, or a portion of an ASP. If the server partition provides the client with a partition of a drive, then the server decides the area of the drive to serve to the client when the network storage space is created.
This allows reads and writes of an ASP to be shared among several virtual devices. If the entire ASP is served to the client, then the rules and procedures apply on the client side as if the drive were local.

Consider the following general performance issues when using virtual SCSI:

- Only use virtual hosted disk in low I/O loads
- Virtual hosted disk is a client/server model, so the combined CPU cycles required on the I/O client and the I/O server will always be higher than local I/O
- If multiple partitions are competing for resources from a virtual hosted disk server, care must be taken to ensure that enough server resources (processor, memory, and disk) are allocated to do the job.
- There is data read caching in memory on the Virtual hosted disk Server partition. Thus, all I/Os that it services could benefit from effects of caching heavily used methods. Read performance can be improved by increasing the memory in the virtual hosted disk server.

6.4 Introduction

In general, applications are functionally isolated from the exact nature of their storage subsystems by the operating system. An application does not have to be aware of whether its storage is contained on one type of disk or another when performing I/O. But different I/O subsystems have subtly different performance qualities, and virtual SCSI is no exception. What differences might an application observe using IBM i operating system Virtual SCSI versus directly attached storage? Broadly, we can categorize the possibilities into I/O latency and I/O bandwidth.

We define *I/O response time* as the time that passes between the initiation of I/O and completion as observed by the application. Latency is a very important attribute of disk I/O. Consider a program that performs 1000 random disk I/Os, one at a time. If the time to complete an average I/O is six milliseconds, the application will run no less than 6 seconds. However, if the average I/O response time is reduced to three milliseconds, the application's run time could be reduced by three seconds. Applications that are multi-threaded or use asynchronous I/O may be less sensitive to latency, but under most circumstances, less latency is better for performance.

We define *I/O bandwidth* as the maximum amount of data that can be read or written to storage in a unit of time. Bandwidth can be measured from a single thread or from a set of threads executing concurrently. Though many applications are more sensitive to latency than bandwidth, bandwidth is crucial for many typical operations such as backup and restore of persistent data.

Because disks are mechanical devices, they tend to be rather slow when compared to high-performance microprocessors such as IBM POWER Systems. As such, we will show that virtual hosted disk performance is comparable to directly attached storage under most workload environments.
IBM i operating system hosts disk space in a Network Storage Space (NWSSTG). A network server description (NWSD) is used to give a name to the configuration, to provide an interface for starting and stopping an AIX logical partition, and to provide a link between AIX and its virtual storage.

There are many factors that affect IBM i operating system performance in a virtual SCSI environment. This chapter discusses some of the common factors and offers guidance on how to help achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and analysis with a primitive disk workload. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics.

All measurements were completed on a POWER5 570+ 4-Way (2.2 GHz). Each system is configured as an LPAR, and each virtual SCSI test was performed between two partitions on the same system with one CPU for each partition. IBM i operating system 5.4 was used on the virtual SCSI server and AIX 5.3 was used on the client partitions.

The primitive disk workload used to evaluative the performance of virtual SCSI is an in house, multi-processed application that performs all types of Synchronous or Asynchronous I/O (read/write/sequential/random) to a target device. The program is run on an AIX or Linux client and gets reports of CPU consumption and gathers disk statistics. Remote statistics are gathered via a socket based application which gathers CPU from the IBM i operating system hosted disk and physical disk statistics.

The purpose of this document is to help virtual SCSI users to better understand the performance of their virtual SCSI system. A customer should be able to size the expected speed of their application from this document.

Note: You will see different terms in this publication that refer to the various components involved with virtual SCSI. Depending on the context, these terms may vary. With SCSI, usually the terms server and client are used, so you may see terms such as virtual SCSI client and virtual SCSI server. On the Hardware Management Console, the terms virtual SCSI server adapter and virtual SCSI client adapter are used. They refer to the same thing. When describing the client/server relationship between the partitions involved in virtual SCSI, the terms hosting partition (meaning the IBM i operating system Server) and hosted partition (meaning the client partition) are used.

6.5 Virtual SCSI Performance Examples

The following sections compare virtual to native I/O performance on bandwidth tests. In these tests, a single thread operates sequentially on a constant file that is 6GB in size, with a dedicated IBM i operating system Server partition. More I/O operations are issued when reading or writing to the file using a small block size than with a larger block size. Because of the larger number of operations and the fact that each operation has a fixed amount of overhead regardless of transfer length, the bandwidth measured with small block sizes is much lower than with large block sizes.

For tests with multiple Network Storage Spaces (NWSS), a thread operates sequentially for each network storage space on a constant file that is 6GB in size, again with a dedicated IBM i operating system Server partition. The following sections compare native vs. virtual, multiple network storage spaces, multiple network storage descriptions, and disk scaling.

6.5.1 Native vs. Virtual Performance

Figure 1 shows a comparison of measured bandwidth using virtual SCSI and local attached DASD for reads with varying block sizes of operations. The difference in the reads between virtual I/O and native I/O in these tests is attributable to the increased latency using virtual I/O. The difference in writes is caused by misalignment, which causes a read for every write. A write alignment change is planned for a future IBM i operating system release which will make virtual and native writes similar in speed.

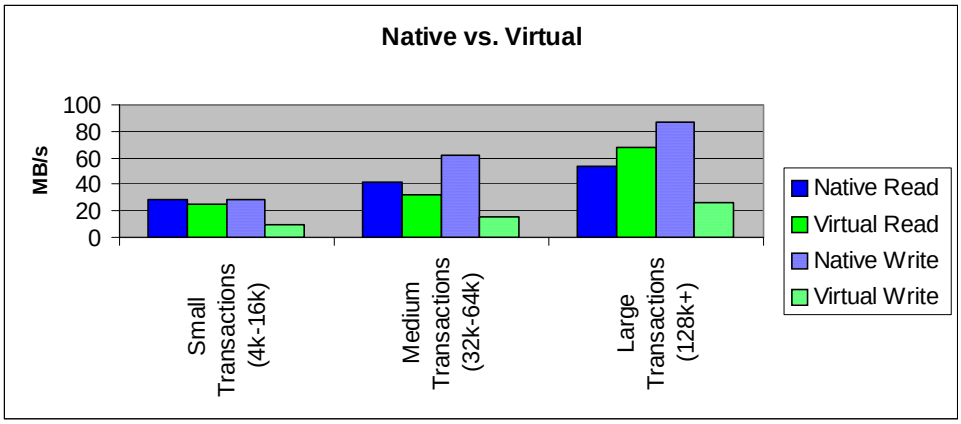


Figure 1 - The figure above shows a comparison of native vs virtual. This experiment shows that virtual write performance is significantly less then native. Read performance performs similar or better then native depending on the read-cache performance.

6.5.2 Virtual SCSI Bandwidth-Multiple Network Storage Spaces

Figure 2 shows a comparison of measured bandwidth while scaling network storage spaces with varying block sizes of operations. The difference in the scaling of these tests is attributable to the performance gain, which can be achieved by adding multiple network storage spaces. This experiment shows that in order to achieve better performance from the hard disk, multiple network storage spaces can be used.

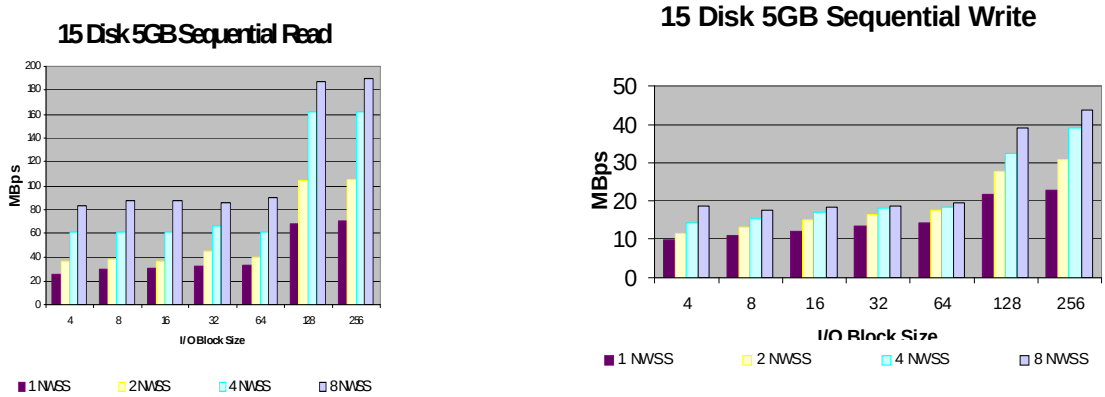


Figure 2- The figures above show performance while scaling network storage spaces. This experiment shows that adding NWSS increases the throughput for read/write performance. The best performance is achieved by using 8 NWSS.

6.5.3 Virtual SCSI Bandwidth-Network Storage Description (NWSD) Scaling

Figure 3 shows a comparison of measured bandwidth while scaling network storage descriptions with varying block sizes of operations. Each of the network storage descriptions have a single network storage space attached to them. The difference in the scaling of these tests is attributable to the performance gain which can be achieved by adding multiple network storage descriptions. This experiment shows that in order to achieve better write performance from the hard disk, multiple network storage descriptions can be used. In order to achieve better performance, 1 network storage space should be used for every 2-4 disk drives in the ASP and each network storage space should be attached to its own network storage description.

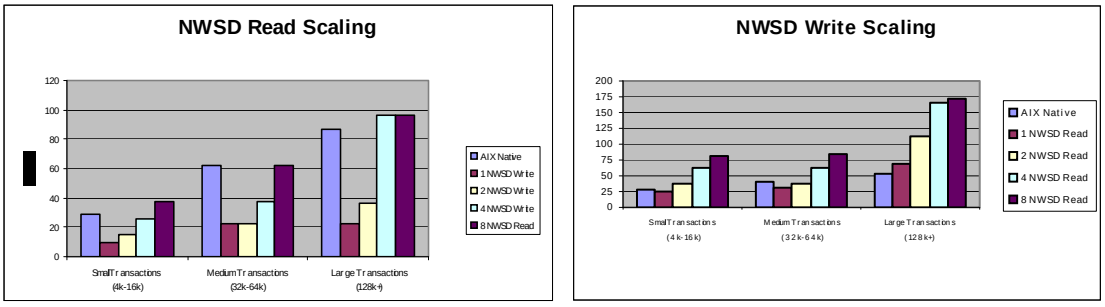


Figure 3 The figures above show performance while scaling network storage descriptions. This experiment shows that adding NWSD increases the throughput for write performance, which was not achievable using 1 network storage description. Read performance increases similar to the network storage space scaling figure.

6.5.4 Virtual SCSI Bandwidth-Disk Scaling

Figure 4 shows a comparison of measured bandwidth while scaling disk drives with varying block sizes of operations. Each of the network storage descriptions have a single network storage space attached to them. The difference in the scaling of these tests is attributable to the performance gain which can be achieved by adding disk drives and IO adapters. The figures below include small (4k-64k) transactions and larger (128k) transactions.

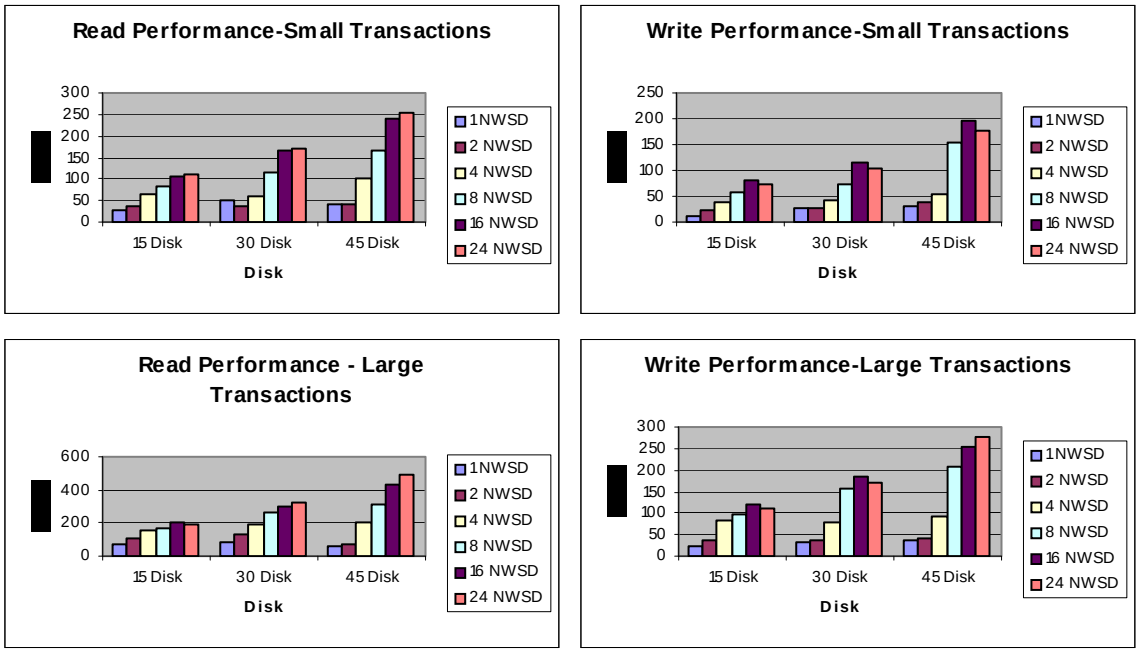


Figure 4 The figures above show read and write performance for small (4k-64k) and large transactions (128k+). This experiment shows that adding disk drives increases the throughput. A system with 45 disk drives will be able to transfer approximately 3 times faster then a system with 15 disk drives. Notice 24-network storage descriptions were used in order to achieve maximum performance.

6.6 Sizing

Sizing methodology is based on the observation that processor time required to perform an I/O on the IBM i operating system Virtual SCSI server is fairly constant for a given I/O size. The I/O devices supported by the Virtual SCSI server are sufficiently similar to provide good recommendations. These numbers are measured at the physical processor.

There are considerations to address when designing and implementing a Virtual SCSI environment. The primary considerations are:

- Dedicated processor server partitions or Micro-Partitioning
- Server partition memory requirements
- One thing that does not have to be factored into sizing is the processor impact of using Virtual I/O on the client. The processor cycles executed on the client to perform a Virtual SCSI I/O are comparable to that of a locally attached I/O. Thus, there is no increase or decrease in sizing on the client partition for a known task.

6.6.1 Sizing when using Dedicated Processors

One sizing method is to size the Virtual SCSI server to the maximum I/O rate of the attached storage subsystem. The sizing could be biased to small I/Os or large I/Os. Sizing to maximum capacity for large I/Os balances the processor capacity of the Virtual SCSI server to the potential I/O bandwidth of the attached I/O. The negative facet of this sizing methodology is that, in nearly every case, we will assign more processor entitlement to the Virtual SCSI server than it typically consumes.

Consider a case where an I/O server manages 15 physical SCSI disks. We can arrive at an upper bound of processors required based on assumptions about the I/O rates that the disks can achieve. If it is known that the workload is dominated by 16 KB operations, we could assume that the 15 disks are capable of 1 read transaction every 36 milliseconds. An IBM i operating system Virtual SCSI server could support around 30,000 read transactions per second on a single processor provided enough disk were present.

To calculate IBM i operating system Virtual SCSI CPU requirements the following formula is provided. The number of transactions per second could be collected by the IBM i operating system command WRKDSKSTS. Based on the average transaction size in WRKDSKSTS, select a number from the table.

		Size of IO						
		4	8	16	32	64	128	256
Type of Transaction	Read	16	22	34	57	92	163	314
	Write	21	26	36	54	82	148	282

Figure 5- CPU milliseconds to process virtual SCSI I/O transaction

The table above shows the time in milliseconds per transaction that Virtual SCSI takes to process one transaction. This value can be used in the formula below to estimate the amount of CPU required per a partition.

$$\frac{(\#\ \text{of Transactions per second} * \text{Time in Milliseconds per transaction})}{1{,}000{,}000} = \text{CPU Utilization}$$

For example.. If your workload performed 10,000 16k read transactions the equation would look like this (34 was selected from the table above):

$$\frac{(10{,}000 * 34)}{1{,}000{,}000} = .34 (34\% \text{ of a total CPU})$$

The total CPU required for a workload, which performs 10,000 16k read transactions per second, would be 34% of a 2.2Ghz POWER5 processor. If a different size processor is used adjust these numbers accordingly. Remember the number chosen in WRKDSKSTS is an average of all I/O's. Your workload could be a mixture of very large transactions and very small transactions. This is to provide a guideline of how to size your CPU correctly, and your results might vary.

Using Dedicated processor partitions may require more CPU then necessary that could be used by other partitions, but will guarantee peak performance. It is most effective if the average I/O size can be estimated so that peak bandwidth does not have to be assumed. Most Virtual SCSI servers will not run at maximum I/O rates all the time, so the use of surplus processor time is potentially wasted by using dedicated processor partitions.

6.6.2 Sizing when using Micro-Partitioning

Defining Virtual SCSI servers in micro-partitions enables much better granularity of processor resource sizing and potential recovery of unused processor time by uncapped partitions. Tempering those benefits, use of micro-partitions for Virtual SCSI servers slightly increases I/O response time and creates somewhat more complex processor entitlement sizing.

The sizing methodology should be based on the same operation costs as for IBM i operating system Server partitions. However, additional entitlement should be added for running in micro-partitions. We recommend that the IBM i operating system Server partition be configured as uncapped so it can take advantage of unused capacity of other partitions, it is possible to get more processor time to service I/O.

Because I/O latency with Virtual SCSI varies with the machine utilization and IBM i operating system Server topology, consider the following:

1. For the most demanding I/O traffic (high bandwidth or very low latency), try to use native I/O.
2. If using Virtual I/O and the system contains enough processors, consider putting the IBM i operating system Server in a dedicated processor partition.
3. If using a Micro-Partitioning IBM i operating system Server, use as few virtual processors as possible.
4. In order to avoid latency issues try to always size the CPU generously.

6.6.3 Sizing memory

The IBM i operating system Virtual SCSI server supports data read caching on the virtual hosted disk server partition. Thus all I/Os that it services could benefit from effects of caching heavily used data. Read performance can vary depending upon the amount of memory which is assigned to the server partition. Workloads which have a small memory footprint can improve their performance greatly by increasing the amount of memory in the IBM i operating system Virtual SCSI server. Alternatively, a system which works on a large amount of data may not see any benefit from caching. The memory for the IBM i operating system Virtual SCSI server in this case can be set at less then 1 GB.

One method to size this is to begin by looking at your ASP in which your network storage space is located. While the system is running the desired workload, type in the command WRKDSKSTS. Write down the average number of I/O request per second in the ASP which is being used by the network storage space. Now dynamically add memory to the partition. Check the number of I/O requests per second once again (remember to reset the statistics using F10). The number of I/O requests per second should lower and your throughput to the IBM i operating system Virtual SCSI server should increase.

Continue adding memory to the IBM i operating system server until you no longer see the number of I/O requests per second change. If your workload changes at a later date the memory can be readjusted accordingly.

Figure 6 below shows a comparison of measured bandwidth of cached transactions with varying block sizes of operations. The figure includes small (4k-64k) transactions and larger(128k) transactions. A partition which runs completely from memory can experience throughput rates as high as 6GB/sec. If it is memory constrained the systems throughput will be lower.

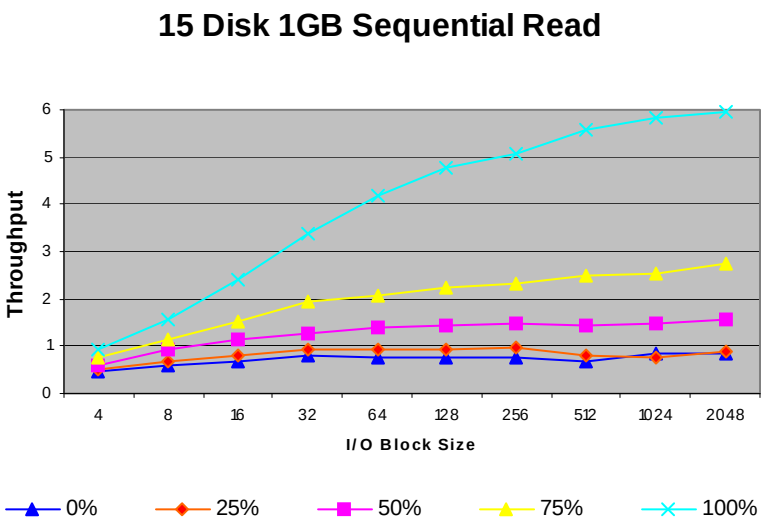


Figure 6 - *The figure above shows a comparison of small, medium, and large transactions affect on memory if cached. The lines represent the amount of data, which is cached in memory. The efficiency of I/O improves with cache hits and larger I/O size. Effectively, there is a fixed latency to start and complete an I/O, with some additional cycle time based on the size of the I/O.*

6.7 AIX Virtual IO Client Performance Guide

The following is a link which will direct you to more in-depth performance tuning for AIX virtual SCSI client:
Advanced POWER Virtualization on IBM p5 Servers: Architecture and Performance Considerations
http://www.redbooks.ibm.com/abstracts/sg247940.html?

6.8 Performance Observations and Tips

- In order to achieve best performance 1 network storage description should be used for every 2-4 disks within an ASP.
- A method to improve write performance is to create 8 NWSD for every 15 disks.
- Best performance was obtained with a network storage description for every network storage space.
- Sizing your memory correctly can improve read performance vastly.
- Multiple network storage descriptions (NWSD) can be attached to a single ASP. No performance benefit from using multiple ASP's was seen.
- For maximum logical volume throughput use multiple network storage spaces attached to a single logical volume.
- With low I/O loads and a small number of partitions, Micro-Partitioning of the IBM i operating system Server partition has little effect on performance.
- For a more efficient Virtual SCSI implementation with larger loads, it may be advantageous to keep the I/O server as a dedicated processor.
- Extensive information can be found at the System i Information Center web site at: http://publib.boulder.ibm.com/iseries.

6.9 Summary

Virtualization is an innovative technology that redefines the utilization and economics of managing an on demand operating environment. POWER5 and future POWER architectures provide new opportunities for clients to take advantage of virtualization capabilities. IBM i operating system family provides the capability for a single physical I/O adapter to be used by multiple logical partitions of the same server, enabling consolidation of I/O resources.

The system resource cost of Virtual SCSI implementation is small, and clients should assess the benefits of the Virtual SCSI implementation for their environment. Simultaneous multithreading should be enabled in a virtual SCSI environment.

Virtual SCSI implementation is an excellent solution for clients looking to consolidate I/O resources with a modest amount of processor. The new IBM i operating system POWER Systems Virtual SCSI capability creates new opportunities for consolidation, and demonstrates strong performance and manageability.

Chapter 7. Logical Partitioning (LPAR)

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

7.1 Introduction

Logical partitioning (LPAR) is a mode of machine operation where multiple copies of operating systems run on a single physical machine.

A *logical partition* is a collection of machine resources that are capable of running an operating system. The resources include processors (and associated caches), main storage, and I/O devices. Partitions operate independently and are logically isolated from other partitions. Communication between partitions is achieved through I/O operations.

The *primary partition* provides functions on which all other partitions are dependent. Any partition that is not a primary partition is a *secondary partition*. A secondary partition can perform an IPL, can be powered off, can dump main storage, and can have PTFs applied independently of the other partitions on the physical machine. The primary partition may affect the secondary partitions when activities occur that cause the primary partition's operation to end. An example is when the PWRDWNSYS command is run on a primary partition. Without the primary partition's continued operation all secondary partitions are ended.

Please refer to the whitepaper 'LPAR Performance on Power Systems with POWER4, POWER5, and POWER6' for the latest information on LPAR performance, located at this site:
http://www.ibm.com/systems/i/advantages/perfmgmt/resource.html

General Tips

- Allocate fractional CPUs wisely. If your sizing indicates two partitions need 0.7 and 0.4 CPUs, see if there will be enough remaining capacity in one of the partitions with 0.6 and 0.4 or else 0.7 and 0.3 CPUs allocated. By adding fractional CPUs up to a "whole" processor, fewer physical processors will be used. Design implies that some performance will be gained.
- Avoid shared processors on large partitions if possible. Since there is a penalty for having shared processors (see later discussion), decide if this is really needed. On a 32 way machine, a whole processor is only about 3 per cent of the configuration. On a 24 way, this is about 4 per cent. Though we haven't measured this, the general penalty for invoking shared processors (often, five per cent) means that rounding up to whole processors may actually gain performance on large machines.

7.2 Considerations

This section provides some guidelines to be used when sizing partitions versus stand-alone systems. The actual results measured on a partitioned system will vary greatly with the workloads used, relative sizes, and how each partition is utilized. For information about CPW values, refer to *Appendix C, "CPW and Relative Performance Values for IBM i"*

When comparing the performance of a standalone system against a single logical partition with similar machine resources, do not expect them to have identical performance values as there is LPAR overhead incurred in managing each partition. For example, consider the measurements we ran on a 4-way system using the standard AS/400 Commercial Processing Workload (CPW) as shown in the chart below.

For the standalone 4-way system we used we measured a CPW value of 1950. We then partitioned the standalone 4-way system into two 2-way partitions. When we added up the partitioned 2-way values as shown below we got a total CPW value of 2044. This is a 5% increase from our measured standalone 4-way CPW value of 1950. I.e. (2044-1950)/1950 = 5 %. The reason for this increased capacity can be attributed primarily to a reduction in the contention for operating system resources that exist on the standalone 4-way system.

Separately, when you compare the CPW values of a standalone 2-way system to one of the partitions (i.e. one of the two 2-ways), you can get a feel for the LPAR overhead cost. Our test measurement showed a capacity degradation of 3%. That is, two standalone 2-ways have a combined CPW value of 2100. The total CPW values of two 2-ways running on a partitioned four way, as shown above, is 2044. I.e. (2100-2044)/2044 = -3%.

The reasons for the LPAR overhead can be attributed to contention for the shared memory bus on a partitioned system, to the aggregate bandwidth of the standalone systems being greater than the bandwidth of the partitioned system, and to a lower number of system resources configured for a system partition than on a standalone system. For example on a standalone 2-way system the main memory available may be X, and on a partitioned system the amount of main storage available for the 2-way partition is X-2.

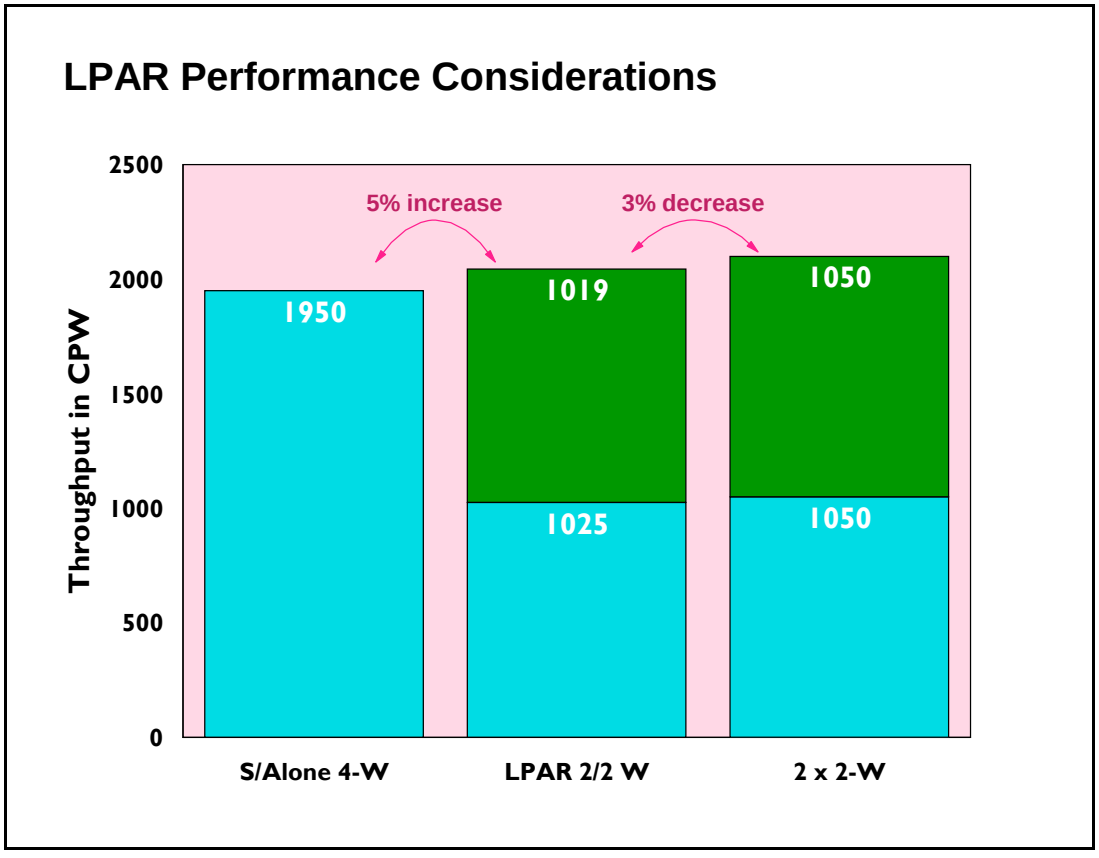


Figure 7.1. LPAR Performance Measured Against Standalone Systems

In summary, the measurements on the 4-way system indicate that when a workload can be logically split between two systems, using LPAR to configure two systems will result in system capacities that are greater than when the two applications are run on a single system, and somewhat less than splitting the applications to run on two physically separate systems. The amount of these differences will vary depending on the size of the system and the nature of the application.

7.3 Performance on a 12-way system

As the machine size increases we have seen an increase in both the performance of a partitioned system and in the LPAR overhead on the partitioned system. As shown below you will notice that the capacity increase and LPAR overhead is greater on a 12-way system than what was shown above on a 4-way system.

Also note that part of the performance increase of an larger system may have come about because of a reduction in contention within the CPW workload itself. That is, the measurement of the standalone 12-way system required a larger number of users to drive the system's CPU to 70 percent than what is required on a 4-way system. The larger number of users may have increased the CPW workload's internal contention. With a lower number of users required to drive the system's CPU to 70 percent on a standalone 4-way system., there is less opportunity for the workload's internal contention to be a factor in the measurements.

The overall performance of a large system depends greatly on the workload and how well the workload scales to the large system. The overall performance of a large partitioned system is far more complicated because the workload of each partition must be considered as well as how each workload scales to the size of the partition and the resources allocated to the partition in which it is running. While the partitions in a system do not contend for the same main storage, processor, or I/O resources, they all use the same main storage bus to access their data. The total contention on the bus affects the performance of each partition, but the degree of impact to each partition depends on its size and workload.

In order to develop guidelines for partitioned systems, the standard AS/400 Commercial Processing Workload (CPW) was run in several environments to better understand two things. First, how does the sum of the capacity of each partition in a system compare to the capacity of that system running as a single image? This is to show the cost of consolidating systems. Second, how does the capacity of a partition compare to that of an equivalently sized stand-alone system?

The experiments were run on a 12-way 740 model with sufficient main storage and DASD arms so that CPU utilization was the key resource. The following data points were collected:

- Stand-alone CPW runs of a 4-way, 6-way, 8-way, and 12-way
- Total CPW capacity of a system partitioned into an 8-way and a 4-way partition
- Total CPW capacity of a system partitioned into two 6-way partitions
- Total CPW capacity of a system partitioned into three 4-way partitions

The total CPW capacity of a partitioned system is greater than the CPW capacity of the stand-alone 12-way, but the percentage increase is inversely proportional to the size of the largest partition. The CPW workload does not scale linearly with the number of processors. The larger the number of processors, the closer the contention on the main storage bus approached the contention level of the stand-alone 12-way system.

For the partition combinations listed above, the total capacity of the 12-way system increases as shown in the chart below.

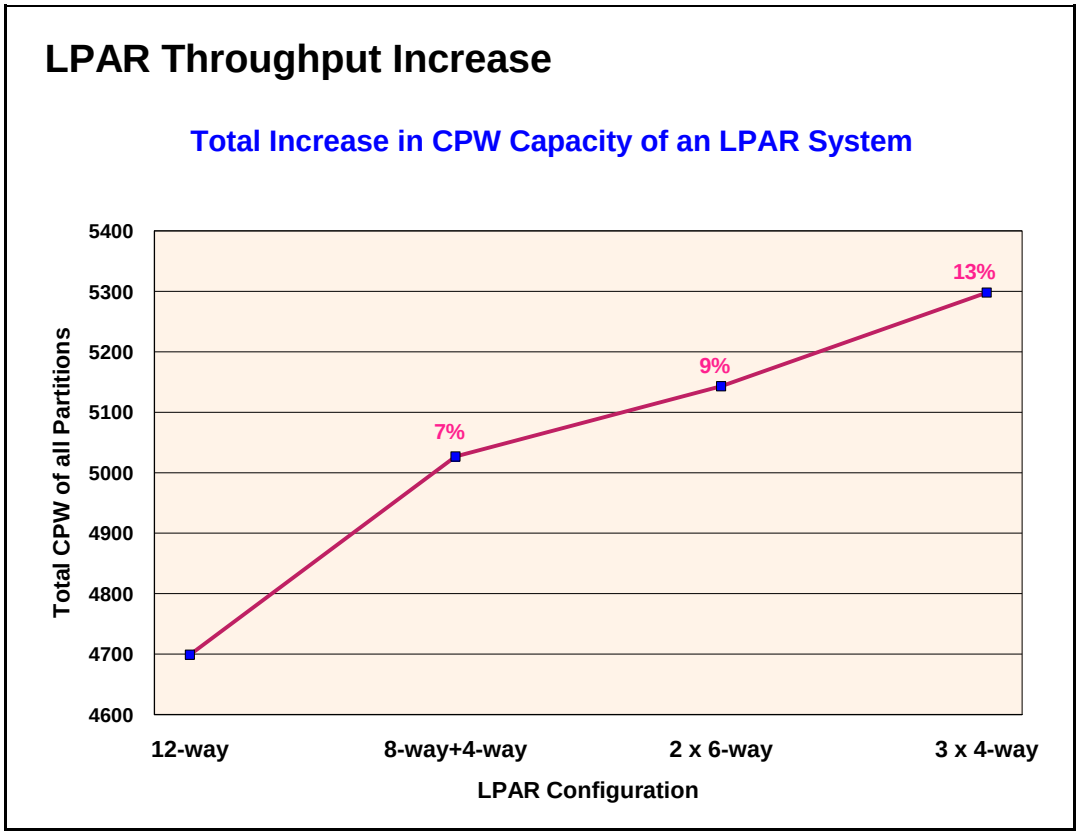


Figure 7.2. 12 way LPAR Throughput Example

To illustrate the impact that varying the workload in the partitions has on an LPAR system, the CPW workload was run at an extremely high utilization in the stand-alone 12-way. This high utilization increased the contention on the main storage bus significantly. This same high utilization CPW benchmark was then run concurrently in the three 4-way partitions. In this environment, the total capacity of the partitioned 12-way exceeded that of the stand-alone 12-way by 18% because the total main storage bus contention of the three 4-way partitions is much less than that of a stand-alone 12-way.

The capacity of a partition of a large system was also compared to the capacity of an equally sized stand-alone system. If all the partitions except the partition running the CPW are idle or at low utilization, the capacity of the partition and an equivalent stand-alone system are nearly identical. However, when all of the partitions of the system were running the CPW, then the total contention for the main storage bus has a measurable effect on each of the partitions.

The impact is greater on the smaller partitions than on the larger partitions because the relative increase of the main storage bus contention is more significant in the smaller partitions. For example, the 4-way partition is degraded by 12% when an 8-way partition is also running the CPW, but the 8-way partition is only degraded by 9%. The two 6-way partitions and three 4-way partitions are all degraded by about 8% when they run CPW together. The impact to each partition is directly proportional to the size of the largest partition.

7.4 Summary

On a partitioned system the capacity increases will range from 5% to 26%. The capacity increase will depend on the number of processors partitioned and on the number of partitions. In general the greater the number of partitions the greater the capacity increase.

When consolidating systems, a reasonable and safe guideline is that a partition may have about 10% less capacity than an equivalent stand-alone system if all partitions will be running their peak loads concurrently. This cross-partition contention is significant enough that the system operator of a partitioned system should consider staggering peak workloads (such as batch windows) as much as possible.

Chapter 8. IPL Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

Performance information for Initial Program Load (IPL) is included in this section.

The primary focus of this section is to present observations from IPL tests on different System i models. The data for both normal and abnormal IPLs are broken down into phases, making it easier to see the detail. For information on previous models see a prior Performance Capabilities Reference.

NOTE: The information that follows is based on performance measurements and analysis done in the Server Group Division laboratory. Actual performance may vary significantly from these tests.

8.1 IPL Performance Considerations

The wide variety of hardware configurations and software environments available make it difficult to characterize a 'typical' IPL environment and predict the results. The following section provides a simple description of the IPL tests.

8.2 IPL Test Description

Normal IPL

- Power On IPL (cold start after managed system was powered down completely)
- For a normal IPL, benchmark time is measured from power-on until the System i server console sign-on screen is available.

Abnormal IPL

- System abnormally terminated causing recovery processing to be done during the IPL. The amount of processing is determined by the system activities at the time the system terminates.
- For an abnormal IPL, the benchmark consists of bringing up a database workload and letting it run until the desired number of jobs are running on the system. Once the workload is stabilized, the system is forced to terminate, forcing a mainstore dump (MSD). The dump is then copied to DASD via the Auto Copy function. The Auto Copy function is enabled through System Service Tools (SST). The System i partition is set to normal so that once the dump is copied, the system completes the remaining IPL with no user intervention. Benchmark time is measured from the time the system is forced to terminate, to when the System i server console sign on screen is available.
- Settings: on the CHGIPLA command the parameter, HDWDIAG, set to (*MIN). All physical files are explicitly journaled. Also logical files are journaled using SMAPP (System Managed Access Path Protection) by using the EDTRCYAP command set to *MIN.

NOTE: Due to some longer starting tasks (like TCP/IP), all workstations may not be up and ready at the same time as the console workstation displays a sign-on screen.

8.3 9406-MMA System Hardware Information

8.3.1 Small system Hardware Configuration

9406-MMA 7051 4 way - 32 GB Mainstore
DASD / 30 70GB 15K rpm arms,
6 DASD in CEC Mirrored
24 DASD in a #5786 EXP24 Disk Drawer attached with a 571F IOA RAID5 Protected

Software Configuration

100,000 spool files (100,000 completed jobs with 1 spool file per job)
500 jobs in job queues (inactive)
600 active jobs in system during Mainstore dump
1000 user profiles, 1000 libraries
Active Database: 2 libraries with 500 physical files and 20 logical files

8.3.2 Large system Hardware Configurations

9406-MMA 7056 8 way - 96 GB Mainstore
DASD / 432 70GB 15K rpm arms,
3 ASP's defined, 108 RAID5 DASD in ASP1, 288 RAID5 DASD in ASP2, 36 DASD no protection in ASP3 - Mainstore dump set to ASP 2

- This system was tested with database files unrelated to this test covering 30% of the DASD space available, this database load causes a long directory recovery.

9406-MMA 7061 16 way - 512 GB Mainstore
DASD / 1000 70GB 15K rpm arms,
3 ASP's defined, 196 Nonconfigured DASD, 120 RAID5 DASD in ASP1, 612 RAID5 DASD in ASP2, 72 DASD no protection in ASP3 - Mainstore dump set to ASP 2

- This system was tested with database files unrelated to this test covering 30% of the DASD space available, this database load causes a long directory recovery.

Software Configuration

400,000 spool files (400,000 completed jobs with 1 spool files each)
1000 jobs waiting on job queues (inactive)
11000 active jobs in system during mainstore dump
200 remote printers, 6000 user profiles, 6000 libraries
Active Database:

- 25 libraries with 2600 physical files and 452 logical files
- 2 libraries with 10,000 physical files and 200 logical files

NOTE:

- Physical files are explicitly journaled
- Logical files are journaled using SMAPP set to *MIN
- Commitment Control used on 20% of the files

8.4 9406-MMA IPL Performance Measurements (Normal)

The following tables provide a comparison summary of the measured performance data for a normal and abnormal IPL. Results provided do not represent any particular customer environment.

Measurement units are in minutes and seconds

Table 8.4.1 iV5R4M5 Normal IPL - Power-On (Cold Start)

	iV5R4M5 GA1 Firmware 4 Way 9406-MMA 7051 32 GB 30 DASD	iV6R1 GA3 Firmware 4 Way 9406-MMA 7051 32 GB 30 DASD	iV5R4M5 GA1 Firmware 8 Way 9406-MMA 7056 96 GB 432 DASD	iV5R4M5 GA1 Firmware 16 Way 9406-MMA 7061 512 GB 1000 DASD	iV6R1 GA3 Firmware 16 Way 9406-MMA 7061 512 GB 1000 DASD
Hardware	**3:10**	**3:12**	**7:53**	**19:17**	**22:07**
SLIC	**4:49**	**5:07**	**7:53**	**10:05**	**9:58**
OS/400	**:48**	**1:23**	**2:12**	**2:41**	**2:22**
Total	**8:47**	**9:42**	**17:58**	**32:03**	**34:27**

Generally, the hardware phase is composed of C1xx xxxx, C3xx xxxx and C7xx xxxx. SLIC is composed of C200 xxxx and C600 xxxx. OS/400 is composed of C900 xxxx SRCs to the System i server console sign-on.

8.5 9406-MMA IPL Performance Measurements (Abnormal)

Measurement units are in minutes and seconds.

Table 8.5.1 iV5R4M5 Abnormal IPL (Partition MSD)

	iV5R4M5 GA1 Firmware 4 Way 9406-MMA 7051 32 GB 30 DASD	iV6R1 GA3 Firmware 4 Way 9406-MMA 7051 32 GB 30 DASD	iV5R4M5 GA1 Firmware 8 Way 9406-MMA 7056 96 GB 432 DASD	iV5R4M5 GA1 Firmware 16 Way 9406-MMA 7061 512 GB 1000 DASD	iV6R1 GA3 Firmware 16 Way 9406-MMA 7061 512 GB 1000 DASD
Processor MSD	**1:50**	**1:02**	**4:12**	**4:28**	**4:34**
SLIC MSD IPL with Copy	**7:23**	**10:45**	**7:00**	**11:35**	**10:56**
Shutdown re-ipl	**2:00**	**2:24**	**3:18**	**2:28**	**3:04**
SLIC re-ipl	**3:09**	**1:29**	**2:32**	**4:02**	**3:28**
OS/400	**4:22**	**5:04**	**28:06**	**29:27**	**20:47**
Total	**18:44**	**20:44**	**45:08**	**52:00**	**42:49**

8.6 NOTES on MSD

MSD is Mainstore Dump. General IPL phase as it relates to the SRCs posted on the operation panel: Processor MSD includes the D2xx xxxx and C2xx xxxx right after the system is forced to terminate. SLIC MSD IPL with Copy follows with the next series of C6xx xxxx, see the next heading for more information on the SLIC MSD IPL with Copy. The copy occurs during the C6xx 4404 SRCs. Shutdown includes the Dxxx xxxx SRCs. Hardware re-ipl includes the next phase of D2xx xxxx and C2xx xxxx. SLIC re-IPL follows which are the C600 xxxx SRCs. OS/400 completes with the C900 xxxx SRCs.

8.6.1 MSD Affects on IPL Performance Measurements

SLIC MSD IPL with Copy is affected by the number of DASD units and the jobs executing at the time of the mainstore dump.

When a system is abnormally terminated, in-process changes to the directories used by the system to manage storage may be lost. During the subsequent IPL, storage management directory recovery is performed to ensure the integrity of the directories and the underlying storage allocations.

The duration of this recovery step will depend on the type of recovery performed and on the size of the directories. In most cases, a subset directory recovery (SRC C6004250) will be performed which may typically run from 2 minutes to 30 minutes depending upon the system. In rare cases, a full directory recovery (SRC C6004260) is performed which typically runs much longer than a subset directory recovery. The duration of the subset directory recovery is dependent on the size of the directory (which relates to the amount of data stored on the system) and on the amount of in-process changes. With the amount of data stored on our largest configurations with one to two thousand disk units, subset directory recovery (SRC C6004250) took from 14 minutes to 50 minutes depending upon the system.

DASD Unit's Effect on MSD Time - Through experimental testing we found the time spent in MSD copying the data to disk is related to the number of DASD arms available. Assigning the MSD copy to an ASP with a larger number of DASD can help reduce your recovery time if an MSD should occur.

8.7 5XX System Hardware Information

8.7.1 5XX Small system Hardware Configuration

520 7457 2 way - 16 GB Mainstore
DASD / 23 35GB 15K rpm arms,
RAID Protected

Software Configuration
100,000 spool files (100,000 completed jobs with 1 spool file per job)
500 jobs in job queues (inactive)
500 active jobs in system during Mainstore dump
1000 user profiles
1000 libraries

Database:
- 2 libraries with 500 physical files and 20 logical files

8.7.2 5XX Large system Hardware Configuration

570 7476 16 way - 256 GB Mainstore
DASD / 924 35GB arms 15K rpm arms,
RAID protected, 3 ASP's defined, majority of the DASD in ASP2 - Mainstore dump was to ASP 2
- This system was tested with 2 TB of database files unrelated to this test, but this load causes a long directory recovery.

595 7499 32-way - 384 GB Mainstore
DASD / 1125 35GB arms 15K rpm arms
RAID protected, 3 ASP's defined, majority of the DASD in ASP2 - Mainstore dump was to ASP 2
- This system was tested with 4 TB of database files unrelated to this test, but this load causes a long directory recovery.

Software Configuration
400,000 spool files (400,000 completed jobs with 1 spool files each)
1000 jobs waiting on job queues (inactive)
11000 active jobs in system during mainstore dump
200 remote printers
6000 user profiles
6000 libraries

Database:
- 25 libraries with 2600 physical files and 452 logical files
- 2 libraries with 10,000 physical files and 200 logical files

NOTE:
- Physical files are explicitly journaled
- Logical files are journaled using SMAPP set to *MIN
- Commitment Control used on 20% of the files

8.8 5XX IPL Performance Measurements (Normal)

The following tables provide a comparison summary of the measured performance data for a normal and abnormal IPL. Results provided do not represent any particular customer environment.

Measurement units are in minutes and seconds

Table 8.8.1 Normal IPL - Power-On (Cold Start)

	V5R3 GA3 Firmware 2 Way 520 7457 16 GB 23 DASD	iV5R4 GA7 Firmware 2 Way 520 7457 16 GB 23 DASD	V5R3 GA3 Firmware 16 Way 570 7476 256 GB 924 DASD	iV5R4 GA7 Firmware 16 Way 570 7476 256 GB 924 DASD	V5R3 GA3 Firmware 32 Way 595 7499 384 GB MS 1125 DASD	iV5R4 GA7 Firmware 32 Way 595 7499 384 GB 1125 DASD
Hardware	5:19	3:30	18:37	17:44	25:50	26:27
SLIC	3:49	4:30	6:42	6:43	8:50	9:36
OS/400	1:00	:50	1:32	2:32	2:30	3:43
Total	10:08	8:50	26:51	26:59	37:10	39:46

The workloads were increased for iV5R4 to better reflect common system load affecting the OS/400 portion of the IPL

Generally, the hardware phase is composed of C1xx xxxx, C3xx xxxx and C7xx xxxx on the 5xx systems. SLIC is composed of C200 xxxx and C600 xxxx. OS/400 is composed of C900 xxxx SRCs to the IBM i operating system console sign-on.

8.9 5XX IPL Performance Measurements (Abnormal)

Measurement units are in hours, minutes and seconds.

Table 8.9.1 Abnormal IPL (Partition MSD)

	V5R3 GA3 Firmware 2 Way 520 7457 16 GB 23 DASD	iV5R4 GA7 Firmware 2 Way 520 7457 16 GB 23 DASD	V5R3 GA3 Firmware 16 Way 570 7476 256 GB 924 DASD	iV5R4 GA7 Firmware 16 Way 570 7476 256 GB 924 DASD	V5R3 GA3 Firmware 32 Way 595 7499 384 GB MS 1125 DASD	iV5R4 GA7 Firmware 32 Way 595 7499 384 GB 1125 DASD
Processor MSD	00:35	4:54	01:53	6:39	02:41	6:06
SLIC MSD IPL with Copy	04:50	15:40	24:10	42:18	43:10	40:03
Shutdown re-ipl	02:46	2:50	04:19	2:23	03:59	3:57
SLIC re-ipl	01:59	2:17	03:59	5:22	04:16	6:21
OS/400	03:21	4:20	09:56	25:45	13:56	44:10
Total	13:31	30:01	44:17	1:22:27	1:08:02	1:40:37

The workloads were increased for iV5R4 to better reflect common system load affecting the MSD and the OS/400 portion of the IPL

8.10 5XX IOP vs IOPLess effects on IPL Performance (Normal)

Measurement units are in minutes and seconds.

Table 8.10.2 Normal IPL - Power-On (Cold Start)

	iV5R4 GA7 Firmware 16 Way IOP 570 7476 256 GB 924 DASD	**iV5R4 GA7 Firmware 16 Way IOPLess 570 7476 256 GB 924 DASD**
Hardware	17:44	18:06
SLIC	6:43	7:20
OS/400	2:32	2:52
Total	26:59	28:18

8.11 IPL Tips

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 following is a partial list of recommendations for IPL performance:

- Remove unnecessary spool files. Use the Display Job Tables (DSPJOBTBL) command to monitor the size of the job table(s) on the system. Change IPL Attributes (CHGIPLA) command can be used to compress job tables if there is a large number of available job table entries. The IPL to compress the tables maybe longer, so try to plan it along with a normal maintenance IPL where you have the time to wait for the table to compress.

- Reduce the number of device descriptions by removing any obsolete device descriptions.

- Control the level of hardware diagnostics by setting the CHGIPLA command to specify HDWDIAG(*MIN), the system will perform only a minimum, critical set of hardware diagnostics. This type of IPL is appropriate in most cases. The exceptions include a suspected hardware problem, or when new hardware, such as additional memory, is being introduced to the system.

- Reduce the amount of rebuild time for access paths during an IPL by using System Managed Access Path Protection (SMAPP). The IBM i operating system Backup and Recovery book (SC41-5304) describes this method for protecting access paths from long recovery times during an IPL.

- For additional information on how to improve IPL performance, refer to *IBM i operating system Basic System Operation, Administration, and Problem Handling (SC41-5206)* - or to the redbook *The System Administrator's Companion to IBM i operating system Availability and Recovery (SG24-2161).*

Chapter 9. Save/Restore Performance

This chapter's focus is on the **IBM i operating system platform.** For legacy system models, older device attachment cards, and the lower performing backup devices see the V5R3 performance capabilities reference.

Many factors influence the observable performance of save and restore operations. These factors include:

- The backup device models, number of DASD units the data is spread across, processors, LPAR configurations, IOA used to attach the devices.
- Workload type: Large Database File, User Mix, Source File, integrated file system
- The use of data compression, data compaction, and Optimum Block Size (USEOPTBLK)
- Directory structure can have a dramatic effect on save and restore operations.

9.1 Supported Backup Device Rates

As you look at backup devices and their performance rates, you need to understand the backup device hardware and the capabilities of that hardware. The different backup devices and IOAs have different capabilities for handling data for the best results in their target market. The following table contains backup devices and rates. Later in this document the rates are used to help determine possible performance. A study of some customer data showed that compaction on their database file data occurred at a ratio of approximately 2.8 to 1. The database files used for the performance workloads were created to simulate that result.

Table 9.1.1 backup device speed and compaction information

Backup Device	Rate (MB/S)	COMPACTION FACTOR
DVD-RAM	2.5	2.8 #1
RDX USB	5.5	2.8
RDX SAS	26	2.8
SLR60	4.0	2.0
SLR100	5.0	2.0
VXA-2	6.0	2.0
6279 VXA-320	12.0	2.0
6258 4MM tape Drive	6.0	2.0
5755 ½ High Ultrium-2	18.0	2.8
3580 Ultrium 2	35.0	2.8
3592-J1a Fiber Channel	40.0	2.8
3580 Ultrium 3 Fiber Channel)	80.0	2.0
5746 Half High Ultrium 4	120.0	2.0
3580 Ultrium 4 Fiber Channel	120.0	2.0
3592-E05 Fiber Channel	100.0	2.5
3580 Ultrium 4 & 5 SAS	120.0	2.0
3592-E07 Fiber Channel	250	2.2
3580 005 Fiber Channel	140	2.8
3580 006 Fiber Channel	160	2.9

#1. Software compression is used here because the hardware doesn't support device compaction
Note the compaction factor is a number that used with the formulas in the following chapter to help describe the actual rates observed as the lab workloads were run using the above drives. This is not the compression ratio of the data being written to tape. I list them here to help understand what our experiments were able to achieve relative to the published drive speed.

9.2 Save Command Parameters that Affect Performance

Use Optimum Block Size (USEOPTBLK)
The USEOPTBLK parameter is used to send a larger block of data to backup devices that can take advantage of the larger block size. Every block of data that is sent has a certain amount of overhead that goes with it. This overhead includes block transfer time, IOA overhead, and backup device overhead. The block size does not change the IOA overhead and backup device overhead, but the number of blocks does. For example, sending 8 small blocks will result in 8 times as much IOA overhead and backup device overhead. This allows the actual transfer time of the data to become the gating factor. In this example, 8 software operations with 8 hardware operations essentially become 1 software operation with 1 hardware operation when USEOPTBLK(*YES) is specified. The usual results are significantly lower CPU utilization and the backup device will perform more efficiently.

Data Compression (DTACPR)
Data compression is the ability to compress strings of identical characters and mark the beginning of the compressed string with a control byte. Strings of blanks from 2 to 63 bytes are compressed to a single byte. Strings of identical characters between 3 and 63 bytes are compressed to 2 bytes. If a string cannot be compressed a control character is still added which will actually expand the data. This parameter is usually used to conserve storage media. If the backup device does not support data compaction, the system i software can be used to compress the data. This situation can require a considerable amount of CPU.

Data Compaction (COMPACT)
Data compaction is the same concept as software compression but available at the hardware level. If you wish to use data compaction, the backup device you choose must support it.

9.3 Workloads

The following workloads were designed to help evaluate the performance of single, concurrent and parallel save and restore operations for selected devices. Familiarization with these workloads can help in understanding differences in the save and restore rates.

Database File related Workloads:
The following workloads are designed to show some possible customer environments using database files.

User Mix **User Mix 3GB, User Mix 12GB -** The User Mix data is contained in a single library and made up of a combination of source files, database files, programs, command objects, data areas, menus, query definitions, etc. User Mix 12GB contains 49,500 objects and User Mix 3GB contains 12,300 objects.

Source File **Source File 1GB -** 96 source files with approximately 30,000 members.

Large Database File **Large File 4GB, 32GB, 64GB**, **320GB -** The Large Database File workload is a single database file. The members in the 4GB and 32GB files are 4GB in size. The Members in the 64GB and 320GB files are 64GB in size.

Integrated File System related Workloads:
Analysis of customer systems indicates about 1.5 to 1 compaction on the tape drives with integrated file system data. This is partly due to the fact that the IBM i operating system programs that store data in the integrated files system, do some disk management functions where they keep the IFS space cleaned up and compressed. And the fact that the objects tend to be smaller by nature, or are mail documents, HTML files or graphic objects that don't compact. The following workloads (1 Directory Many Objects, Many Directories Many Objects, Domino, Network Storage Space) show some possible customer integrated file system environments.

1 Directory Many objects This integrated file system workload consists of 111,111 stream files in a single directory where the stream files have 32K of allocated space, 24K of which is data. Approximately 4 GB total sampling size.

Many Directories Many objects This integrated file system workload is 6 levels deep, 10 directories wide where each directory level contains 10 directories resulting in a total of 111,111 Directories and 111,111 stream files, where the stream files have 32K of allocated space, 24K of which is data. Approximately 5 GB total sampling size.

Domino This integrated file system workload consists of a single directory containing 90 mail files. Each mail file is 152 MB in size. The mail files contain mail documents with attachments where approximately 75% of the 152 MB is attachments. Approximately 13 GB total sampling size.

Network Storage Space This integrated file system workload consists of a Linux storage space of approximately 6 GB total sampling size.

9.4 Comparing Performance Data

When comparing the performance data in this document with the actual performance on your system, remember that the performance of save and restore operations is data dependent. If the same backup device was used on data from three different systems, three different rates may result.

The performance of save and restore operations are also dependent on the system configuration, most directly affected by the number and type of DASD units on which the data is stored and by the type of storage IOAs being used.

Generally speaking, the Large Database File data that was used in testing for this document was designed to compact at an approximate 2.8 to1 ratio. If we were to write a formula to illustrate how performance ratings are obtained, it would be as follows:

((DeviceSpeed * LossFromWorkLoadType) * Compaction) = MB/Sec * 3600 = MB/HR /1000 = GB/HR.

But the reality of this formula is that the "LossFromWorkLoadType" is far more complex than described here. The different workloads have different overheads, different compaction rates, and the backup devices use different buffer sizes and different compaction algorithms. The attempt here is to group these

workloads as examples of what might happen with a certain type of backup device and a certain workload.

Note: Remember that these formulas and charts are to give you an idea of what you might achieve from a particular backup device. Your data is as unique as your company and the correct backup device solution must take into account many different factors.

The save and restore rates listed in this document were obtained on a dedicated system. A dedicated system is one where the system is up and fully functioning but no other users or jobs are running except the save and restore operations. All processors and Memory were dedicated to the system and no partial processors were used. Other subsystems such as QBATCH are required in order to run concurrent and parallel operations. All workloads were deleted before restoring them again.

9.5 Lower Performing Backup Devices

With the lower performing backup devices, the devices themselves become the gating factor so the save rates are approximately the same, regardless of system CPU size (DVD-RAM).

Table 9.5.1 Lower performing backup devices LossFromWorkLoadType Approximations (Save Operations)

Workload Type	Amount of Loss
Large Database File	95%
User Mix / Domino / Network Storage Space	55%
Source File / 1 Directory Many Objects / Many Directories Many Objects	25%

Example for a DVD-RAM:

DeviceSpeed * LossFromWorkLoad * Compaction Factor

0.75 * 0.95 = (.71) * 2.8 = (1.995) MB/S * 3600 = 7182 MB/HR = 7 GB/HR

0.75 * 0.95 = (.71) * No Compression * 3600 = 2556 MB/HR = 2.5 GB/HR

9.6 Medium & High Performing Backup Devices

Medium & high performing backup devices (SLR60, SLR100, VXA-2, VXA-320).

Table 9.6.1 Medium performing backup devices LossFromWorkLoadType Approximations (Save Operations)

Workload Type	Amount of Loss
Large Database File	95%
User Mix / Domino / Network Storage Space	65%
Source File / 1 Directory Many Objects / Many Directories Many Objects	25%

Example for SLR100:

DeviceSpeed * LossFromWorkLoad * Compaction Factor

5.0 * 0.95 = (4.75) * 2.0 = (9.5) MB/S * 3600 = 34200 MB/HR = 34 GB/HR

9.7 Ultra High Performing Backup Devices

High speed backup devices are designed to perform best on large files. The use of multiple high speed backup devices concurrently or in parallel can also help to minimize system save times. See section on Multiple backup devices for more information (3580 Ultrium-2, 3580 Ultrium-3 (2Gb & 4Gb Fiber Channel), 3592-J1a).

Table 9.7.1 Higher performing backup devices LossFromWorkLoadType Approximations (Save Operations)	
Workload Type	**Amount of Loss**
Large Database File	95%
User Mix / Domino / Network Storage Space	50%
Source File / 1 Directory Many Objects / Many Directories Many Objects	5%

Example for 3580 ULTRIUM-2 Fiber:

DeviceSpeed * LossFromWorkLoad * Compaction Factor

LG File 35.0 * 0.95 = (33.25) * 2.8= (93.1) MB/S *3600 = 335160 MB/HR = 335 GB/HR

UserMix 35.0 * 0.50 = (17.5) * 2.8= (49) MB/S *3600 = 176400 MB/HR = 176 GB/HR

Source 35.0 * 0.05 = (1.75) * 2.8= (4.9) MB/S *3600 = 17640 MB/HR = 17.6 GB/HR

NOTE: Actual performance is data dependent, these formulas are for estimating purposes and may not match actual performance on customer systems.

9.8 The Use of Multiple Backup Devices

Concurrent Saves and Restores - The ability to save or restore different objects from a single library/directory to multiple backup devices or different libraries/directories to multiple backup devices at the **same time** from **different jobs**. The workloads that were used for the testing were Large Database File and User Mix from libraries. For the tests multiple identical libraries were created, a library for each backup device being used.

Parallel Saves and Restores - The ability to save or restore a **single object** or library/directory across **multiple backup devices** from the **same job**. Understand that the function was designed to help those customers, with very large database files which are dominating the backup window. The goal is to provide them with options to help reduce that window. Large objects, using multiple backup devices, using the parallel function, can greatly reduce the time needed for the object operation to complete as compared to a serial operation on the same object.

Concurrent operations to multiple backup devices will probably be the preferred solution for most customers. The customers will have to weigh the benefits of using parallel verses concurrent operations for multiple backup devices in their environment. The following are some thoughts on possible solutions to save and restore situations. Remember that memory, processors and DASD play a large factor in whether or not you will be able to make use of parallel or concurrent operations that can be used to affect the back up window.

- For save and restore with a User Mix or small to medium object workloads, the use of concurrent operations will allow multiple objects to be processed at the same time from different jobs, making better use of the backup devices and the system.

- For systems with a large quantity of data and a few very large database files whether in libraries or directories, a mixture of concurrent and parallel might be helpful. (Example: Save all of the libraries/directories to one backup device, omitting the large files from the library or the directory the file is located in. At the same time run a parallel save of those large files to multiple backup devices.)

- For systems dominated by Large Files the only way to make use of multiple backup devices is by using the parallel function.

- For systems with a few very large files that can be balanced over the backup devices, use concurrent saves.

- For backups where libraries/directories increase or decrease in size significantly throwing concurrent saves out of balance constantly, the customer might benefit from the parallel function as the libraries/directories would tend to be balanced against the backup devices no matter how the libraries change. Again this depends upon the size and number of data objects on the system.

- Customers planning for future growth where they would be adding backup devices over time, might benefit by being able to set up Backup Recovery Media Services (BRMS/400) using *AVAIL for backup devices. Then when a new backup device is added to the system and recognized by BRMS/400 it will be used, leaving the BRMS/400 configuration the same but benefiting from the additional backup device. Also the same is true in reverse: If a backup device is lost, the weekly backup doesn't have to be postponed and the BRMS/400 configuration doesn't need to change, the backup will just use the available backup devices at the time of the save.

9.9 Parallel and Concurrent Library Measurements

This section discusses parallel and concurrent library measurements for tape drives, while sections later in this chapter discuss measurements for virtual tape drives.

9.9.1 Hardware (2757 IOAs, 2844 IOPs, 15K RPM DASD)

Hardware Environment.
This testing consisted of an 840 24 way system with 128 GB of memory. The model 840 doesn't support the 15K RPM DASD in the main tower so only 4, 18 GB 10K RPM RAID protected DASD units were in the main tower.

15 PCI-X towers (5094 towers), were attached and filled with 45, 35 GB 15K RPM RAID protected DASD units. 2757 IOAs in all 15 towers and 2844 IOPs. All of the towers attached to the system were configured into 8 High Speed Link (HSL) with two towers in each link. One 5704 fiber channel connector in each tower, or two per HSL. A total of 679 DASD, 675 of which were 35 GB 15K RPM DASD units all in the system ASP. We used the new high speed ULTRIUM GEN 2 tape drives, model 3580 002 fiber channel attached.

There were a lot of different options we could have chosen to try to view this new hardware, we were looking for a reasonable system to get the maximum data flow, knowing that at some point someone will ask what is the maximum. As you look at this information you will need to put it in prospective of your own system or system needs.

We chose 8 HSLs because our bus information would tell us that we can only flow so much data across a single HSL. The total number of 3580 002 tape drives we believe we could put on a link was something a little greater than 2, but the 3rd tape drive would probably be slowed greatly by what the HSL could support, so to maximize the data flow we chose to put only two on a HSL.

What does this mean to your configuration? If you are running large file save and restore operations we would recommend only 2 high speed tape drives per HSL. If your data leans more toward user mix you

could probably make use of more drives in a single HSL. How many will depend upon your data. Remember there are other factors that affect save and restore operations, like memory, number of processors available, number and type of DASD available to feed those tape drives, and type of storage IOAs being used.

Large File operations create a great deal of data flow without using a lot of processing power but User Mix data will need those Processors, memory and DASD. Could the large file tests have been done by fewer processors? Yes, probably by something between 8 and 16 but in order to also do the user mix in the same environment we choose to have the 24 processors available. The user mix is a more generic customer environment and will be informational to a larger set of customers and we wanted to be able to provide some comparison information for most customers to be able to use here.

9.9.2 Large File Concurrent

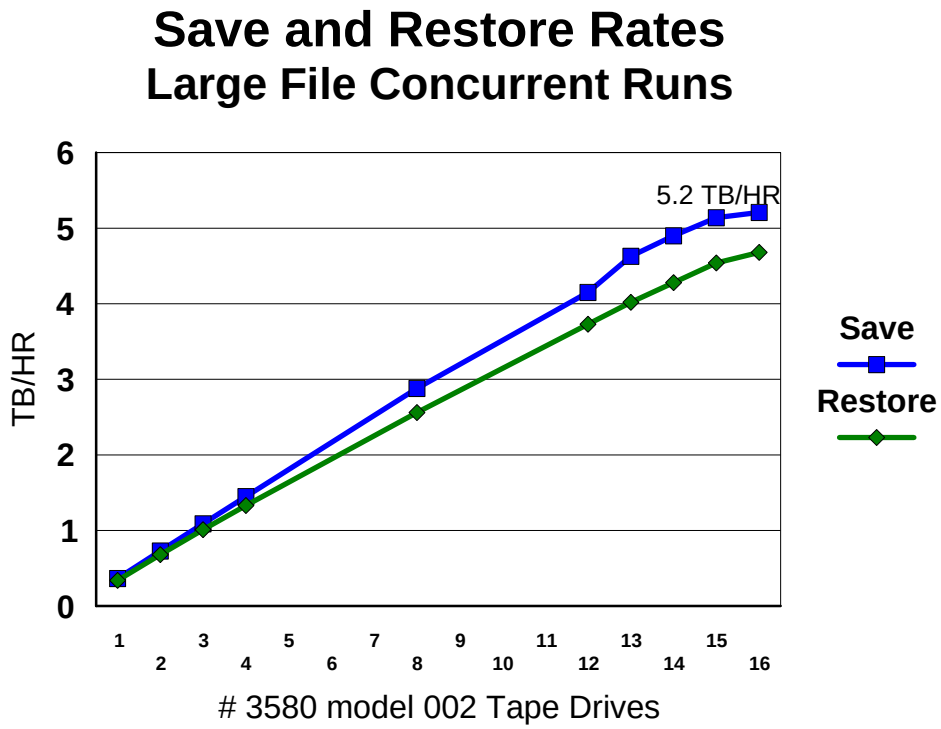
For the concurrent testing 16 libraries were built, each containing a single 320 GB file with 80 4 GB members. The file size was chosen to sustain a flow across the HSL, system bus, processors, memory and tapes drives for about an hour. We were not interested in peak performance here but sustained performance. Measurements were done to show scaling from 1 to 16 tape drives, knowing that near the top number of tape drives that the system would become the limiting factor and not the tape drives. This could be used by customers to give them an estimate at what might be a reasonable number of tape drives for their situation.

Table 9.9.2.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Concurrent) (Save = S, & Restore = R)

# 3580.002 Tape drives		1	2	3	4		8		12	13	14	15	16
320 GB DB file with 80 4 GB members	S	365 GB/HR	730 GB/HR	1.09 TB/HR	1.45 TB/HR		2.88 TB/HR		4.15 TB/HR	4.63 TB/HR	4.90 TB/HR	5.14 TB/HR	5.21 TB/HR
	R	340 GB/HR	680 GB/HR	1.01 TB/HR	1.33 TB/HR		2.56 TB/HR		3.73 TB/HR	4.02 TB/HR	4.28 TB/HR	4.54 TB/HR	4.68 TB/HR

In the table above, you will notice that the 16th drive starts to loose value. Even though there is gain we feel we are starting to see the system saturation points start to factor in. Unfortunately, we didn't have

anymore drives to add in but believe that the total data throughput would be relatively equal, even if any more drives were added.

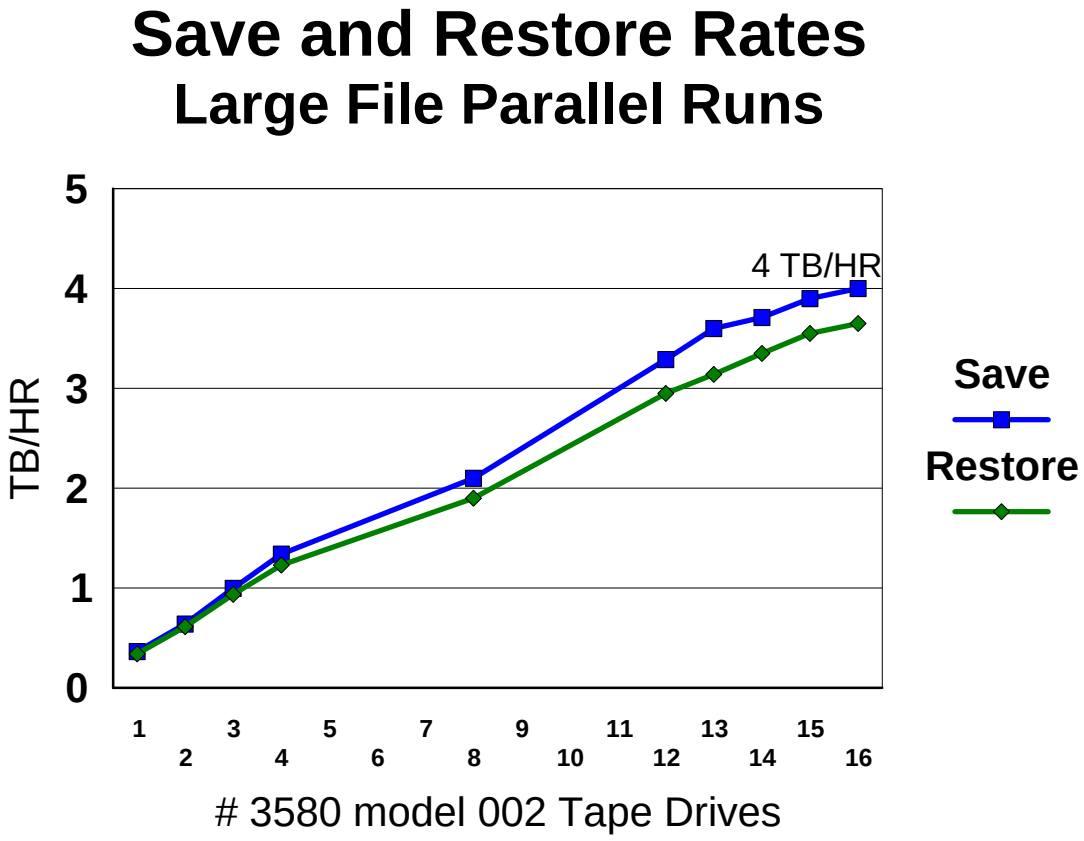


9.9.3 Large File Parallel

For the measurements in this environment, BRMS was used to manage the save and restore, taking advantage of the ability built into BRMS to split an object between multiple tape drives. Starting with a 320 GB file in a single library and building it up to 2.1 TB for tape drive tests 1 - 4 and 8. The file was then duplicated in the library for tape drive tests 12 - 16, a single library with two 2.1 TB files was used. Not quite the same as having a 4.2 TB file. Because of certain limitations in building our test data, we felt this was the best way to build the test data. The goal is to see scaling of tape drives on the system along with trying to locate any saturation points that might help our customers identify limitations in their own environment.

Table 9.9.3.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Parallel)
(Save = S, & Restore = R)

# 3580.002 Tape drives	1	2	3	4		8		12	13	14	15	16
S	363 GB/HR	641 GB/HR	997 GB/HR	1.34 TB/HR		2.1 TB/HR		3.29 TB/HR	3.60 TB/HR	3.71 TB/HR	3.90 TB/HR	4 TB/HR
R	340 GB/HR	613 GB/HR	936 GB/HR	1.23 TB/HR		1.90 TB/HR		2.95 TB/HR	3.14 TB/HR	3.35 TB/HR	3.55 TB/HR	3.65 TB/HR

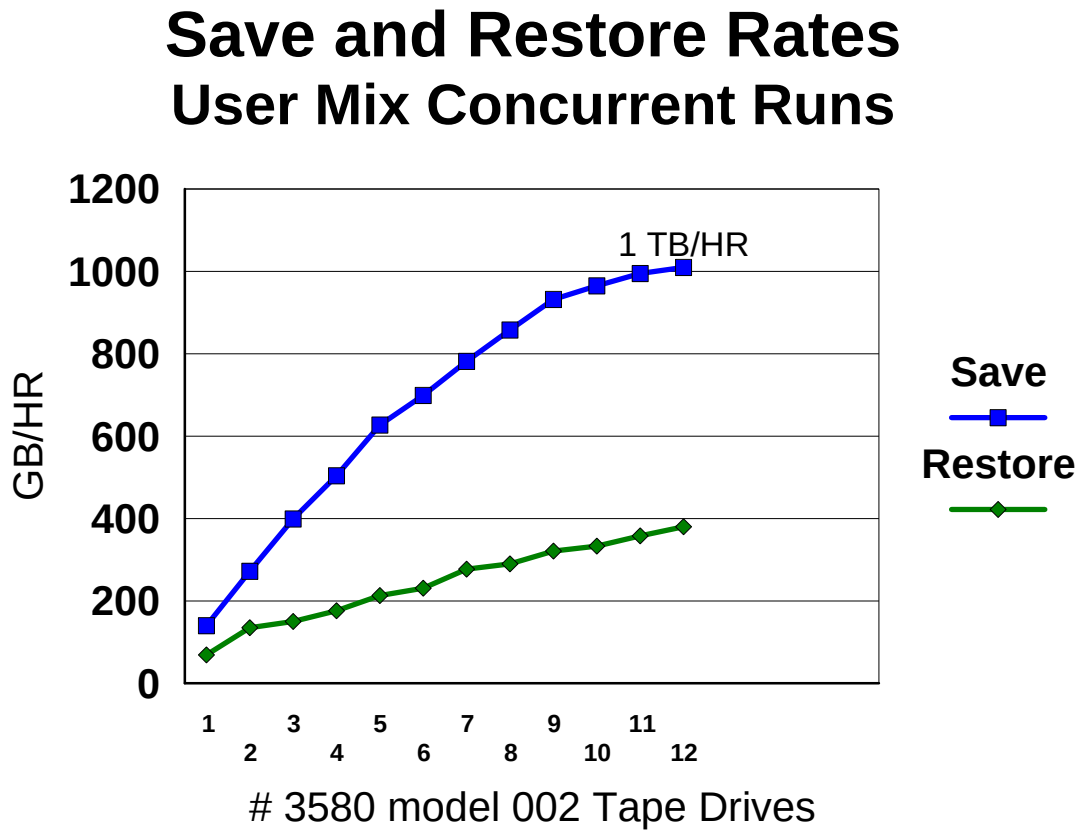


9.9.4 User Mix Concurrent

User Mix will generally portray a fair population of customer systems, where the real data is a mixture of programs, menus, commands along with their database files. The new ultra tape drives are in their glory when streaming large file data, but a lot of other factors play a part when saving and restoring multiple smaller objects.

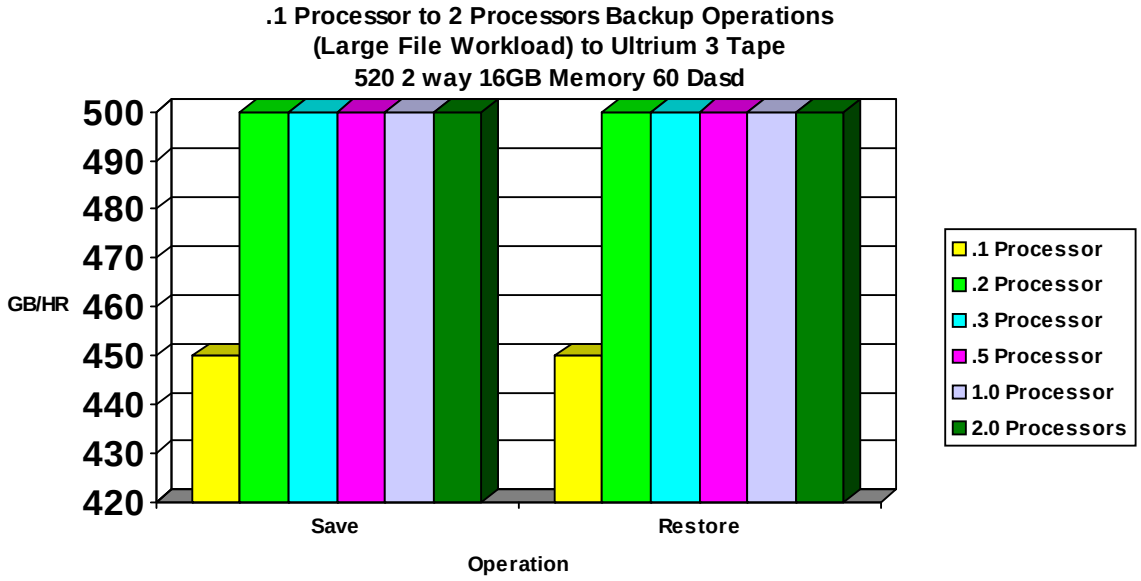
Table 9.9.4.1 iV5R2 16 - 3580.002 Fiber Channel Tape Device Measurements (Concurrent) (Save = S, & Restore = R)

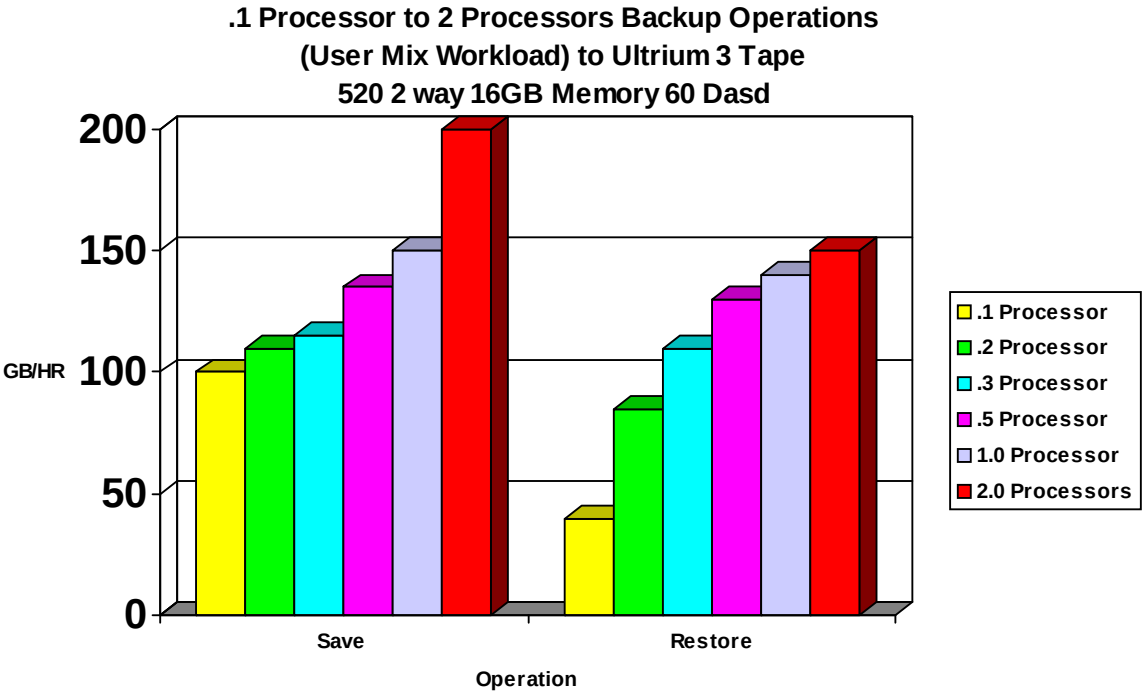
# 3580.002 Tape drives		1	2	3	4	5	6	7	8	9	10	11	12
12 GB total Library size workload was used for modeling this, as described in section 9.3	S	140 GB/HR	272 GB/HR	399 GB/HR	504 GB/HR	627 GB/HR	699 GB/HR	782 GB/HR	858 GB/HR	932 GB/HR	965 GB/HR	995 GB/HR	1010 GB/HR
	R	69 GB/HR	135 GB/HR	150 GB/HR	176 GB/HR	213 GB/HR	231 GB/HR	277 GB/HR	290 GB/HR	321 GB/HR	333 GB/HR	358 GB/HR	380 GB/HR



9.10 Number of Processors Affect Performance

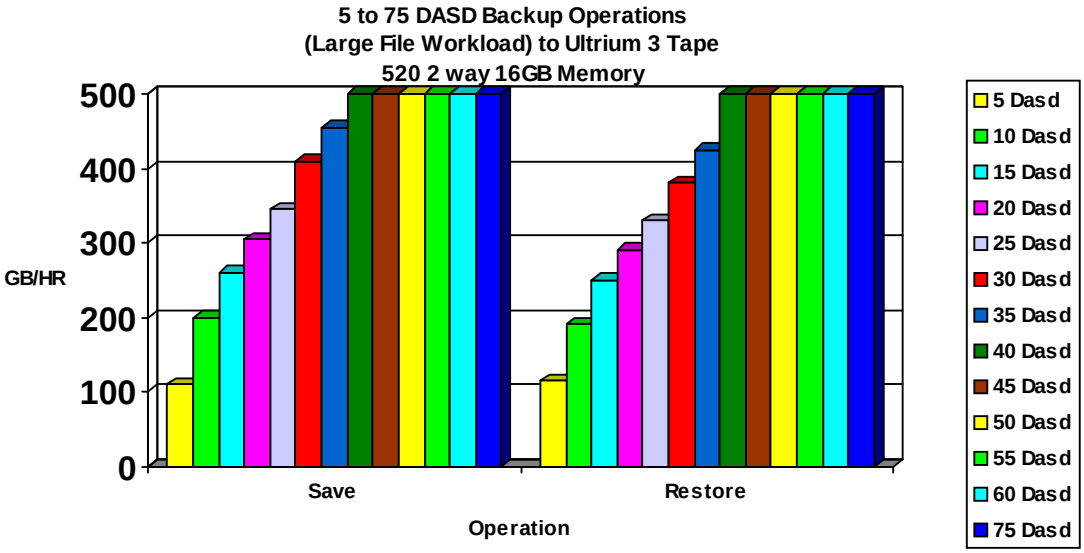
With the Large Database File workload, it is possible to fully feed two backup devices with a single processor, but with the User Mix workload it takes 1+ processors to fully feed a backup device. A recommendation might be 1 and 1/3 processors for each backup device you want to feed with User Mix data.

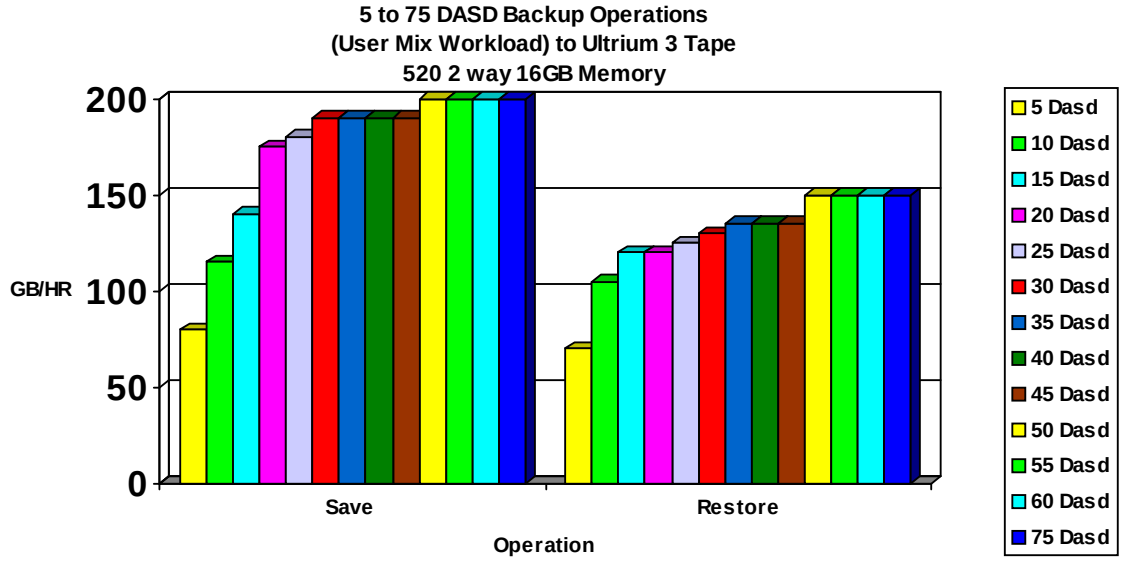




9.11 DASD and Backup Devices Sharing a Tower

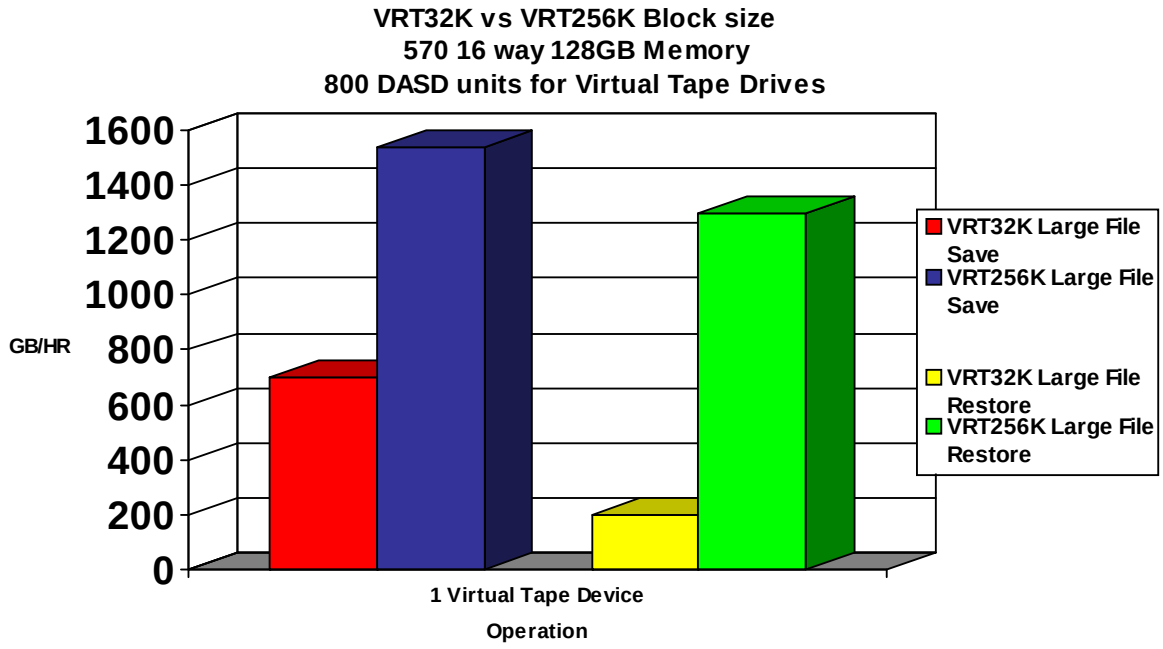
The system architecture does not require that DASD and backup devices be kept separated. Testing in the IBM Rochester Lab, we had attached one backup device to each tower and all towers had 45 DASD units in them, when we did the 3580 002 testing. The 3592-J1a has similar characteristics to the 3580 002 but the 3580 003 and 3592-E05 models have greater capacities which create new scenarios. You aren't physically limited to putting one backup device in a tower, but for the newest high speed backup devices you can saturate the bus if you have multiple devices in a tower. You need to look at your total system or partition configuration in order to determine if it is possible to use multiple high speed devices on the system and still get the most out of these devices. No matter what you determine is possible we advocate spreading your backup devices amongst the towers available.



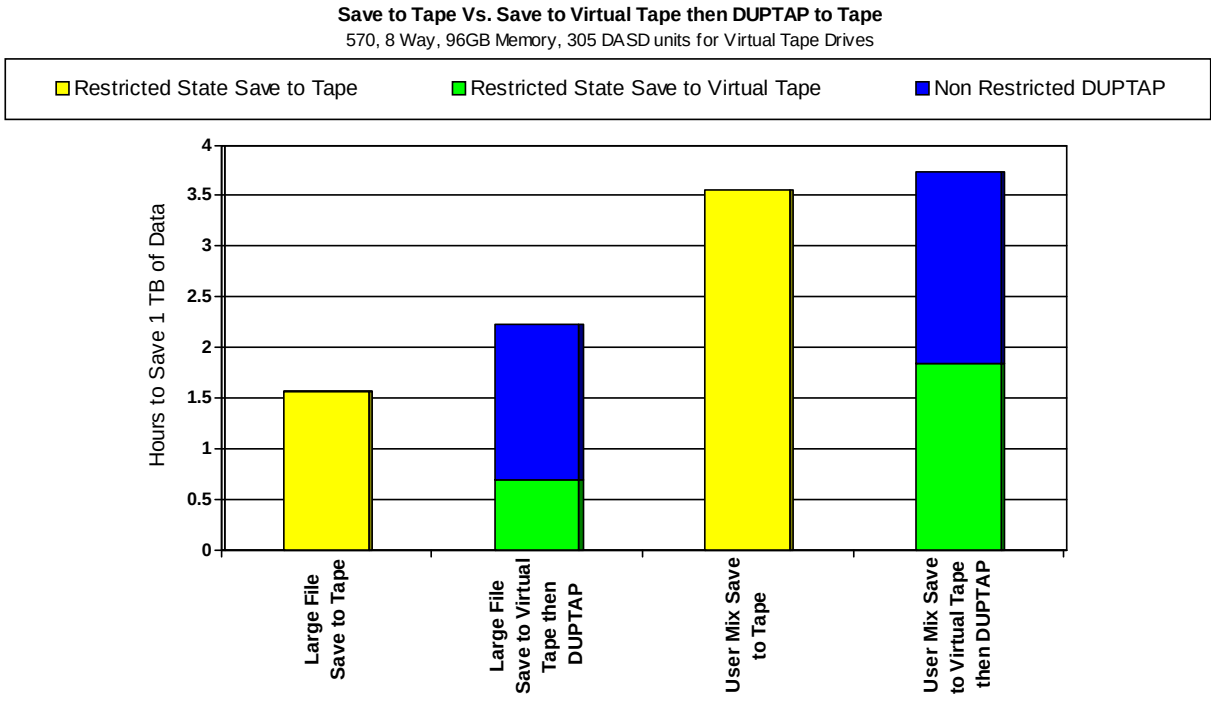


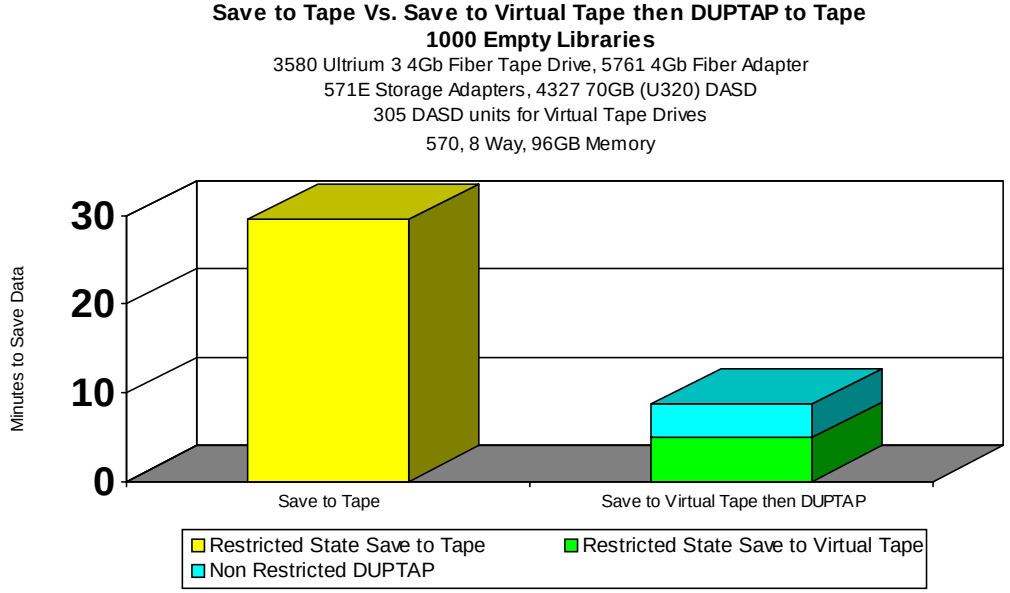
9.12 Virtual Tape

Virtual tape drives are being introduced in iV5R4 so those customers can make use of the speed of saving to DASD, then save the data using DUPTAP to the tape drives reducing the backup window where the system is unavailable to users. There are a lot of pieces to consider in setting up and using Virtual tape drives. The block size must match the physical backup device block capabilities you will be using. The following helps to show that even if your workload is large file you may not gain anything in your back up window even using the virtual tape drives. If your tape drive uses smaller block sizes your virtual tape drive must use small blocks



The following measurements were done on a system with newer hardware including a 3580 Ultrium 3 4Gb Fiber Channel Tape Drive, 571E storage adapters, and 4327 70GB (U320) DASD.

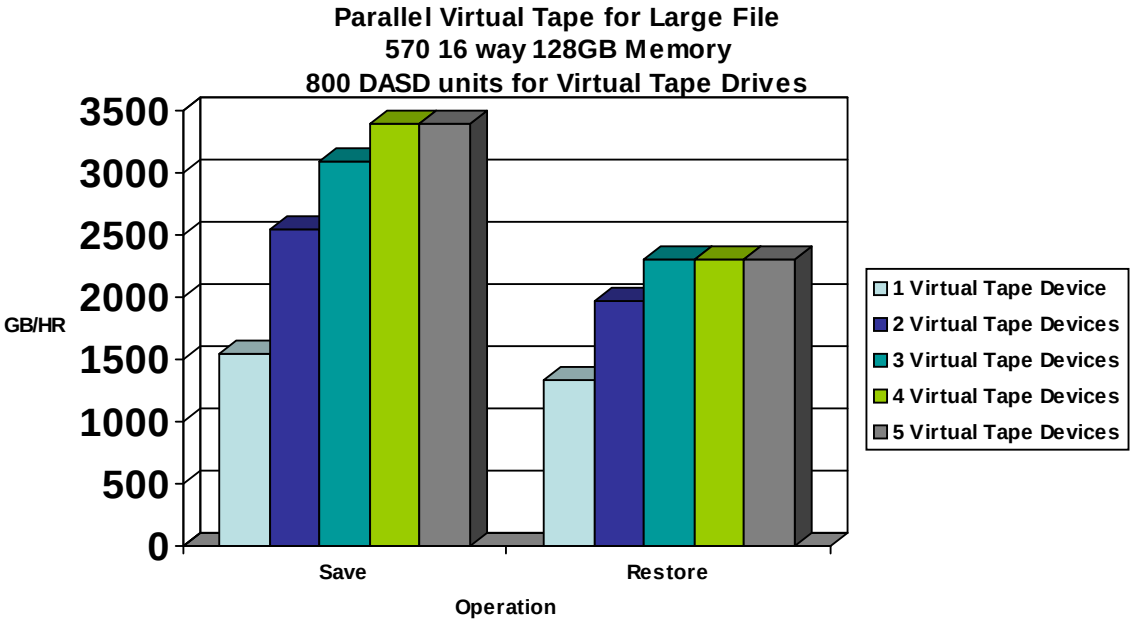


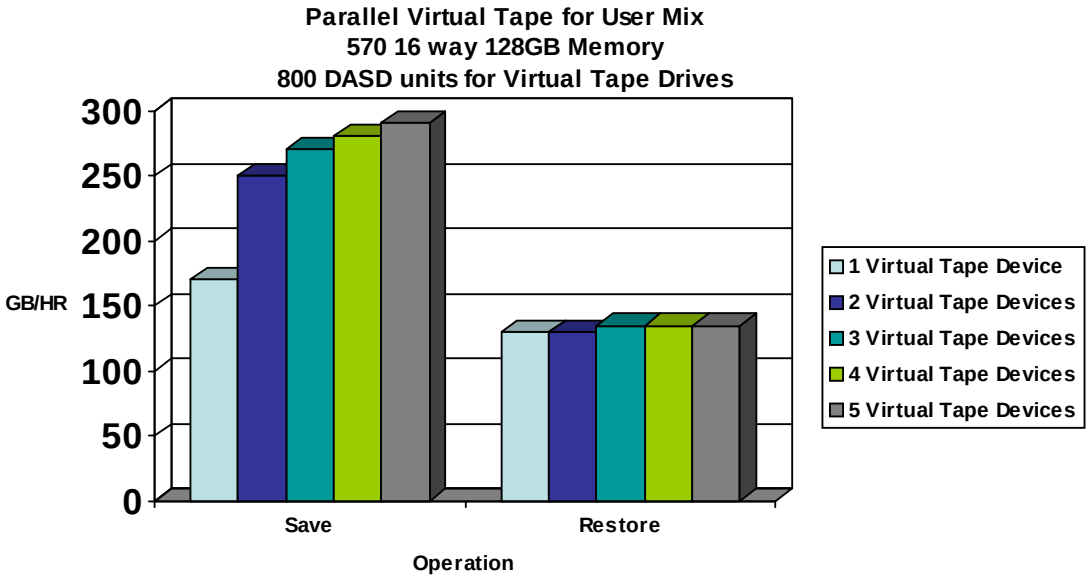


Measurements were also done comparing save of 1000 empty libraries to tape versus save of these libraries to virtual tape followed by DUPTAP from the virtual tape to tape. The save to tape was much slower which can be explained as follows. When data is being saved to tape, a flush buffer is requested after each file is written to ensure that the file is actually on the tape. This forces the drive to backhitch for each file and greatly reduces the performance. The DUPTAP command does not need to send a flush buffer until the duplicate command completes, so it does not have the same performance impact.

9.13 Parallel Virtual Tapes

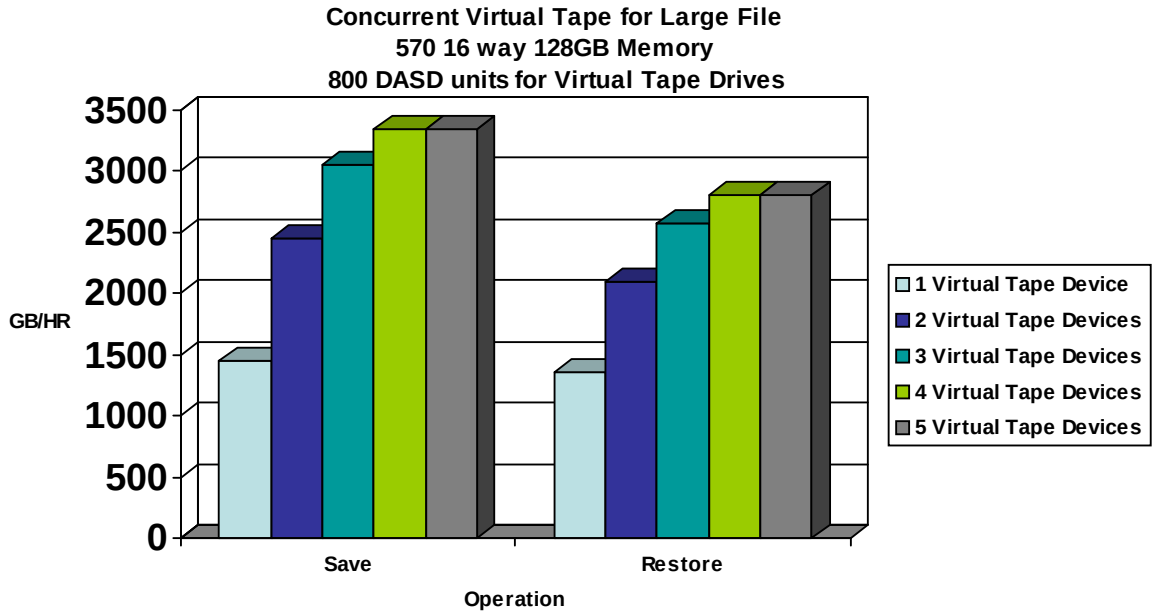
NOTE: Virtual tape is reading and writing to the same DASD so the maximum throughput with our concurrent and parallel measurements is different than our tape drive tests where we were reading from DASD and writing to tape.

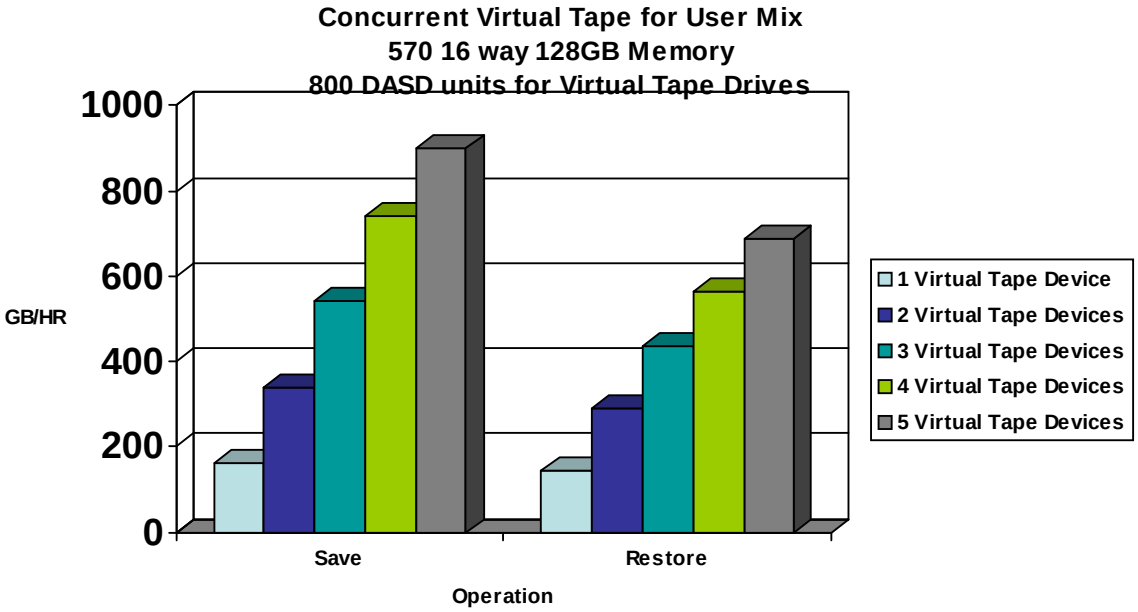




9.14 Concurrent Virtual Tapes

NOTE: Virtual tape is reading and writing to the same DASD so the maximum throughput with our concurrent and parallel measurements is different than our tape drive tests where we were reading from DASD and writing to tape.

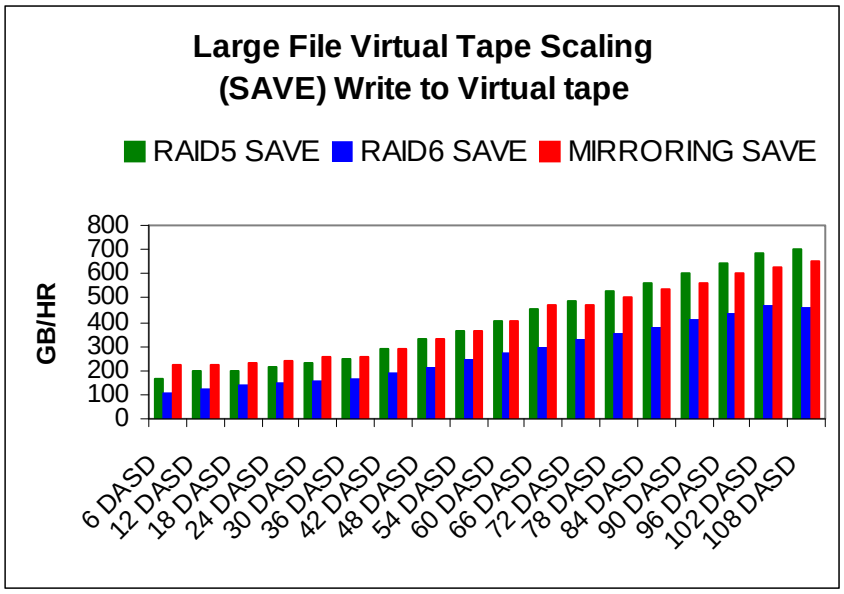


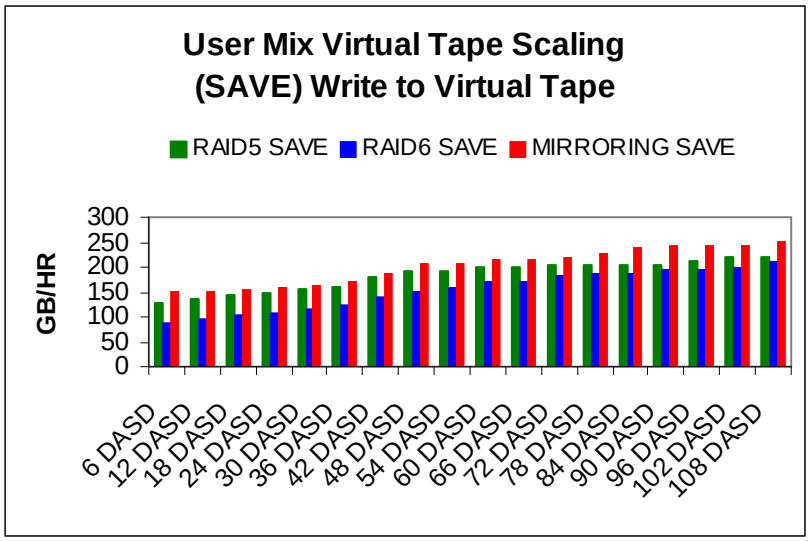


9.15 Save and Restore Scaling using a Virtual Tape Drive

A 570 8 way System i was used for the following tests. A user ASP was created using up to 3 571F IOAs with up to 36 U320 70 GB DASD on each IOA. The Chart shows the number of DASD in each test and the Virtual tape drive was created using that DASD.

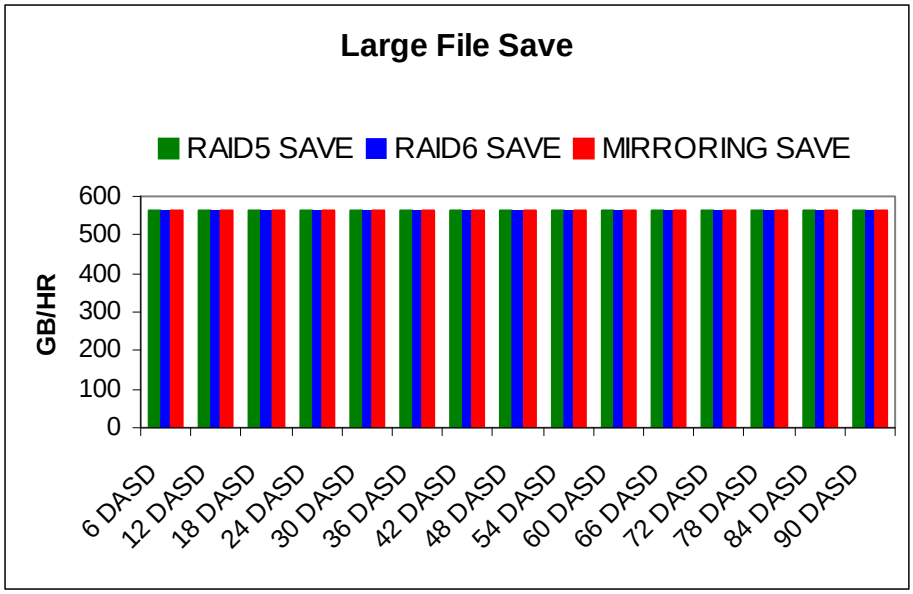
The workload data was restored into the system ASP and was then saved to the Virtual tape drive in the user ASP. The system ASP consisted of 2 HSL loops, a mix of 571E and 571F IOAs and 312 - 70GB U320 DASD units. These charts are very specific to this DASD but the scaling flow would be similar with different IOAs the actual rates would vary. For more information on the IOAs and DASD see Chapter 4 of this guide. Restoring the workloads from the Virtual tape drives started at 900 GB/HR reading from 6 DASD and scaled up to 1.5 TB/HR on the 108 DASD. The bottle neck will be limited to where you are writing and how many DASD are available to the write operation.

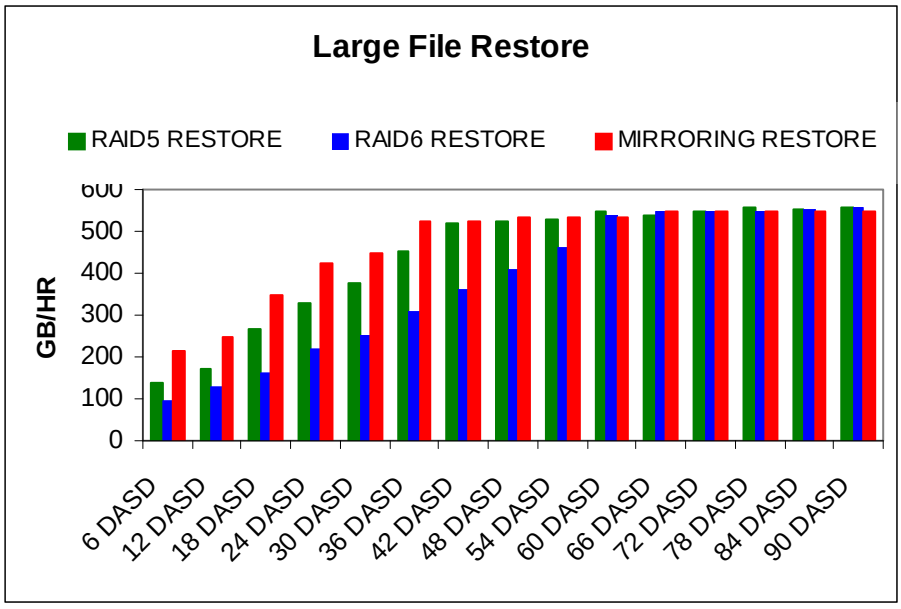


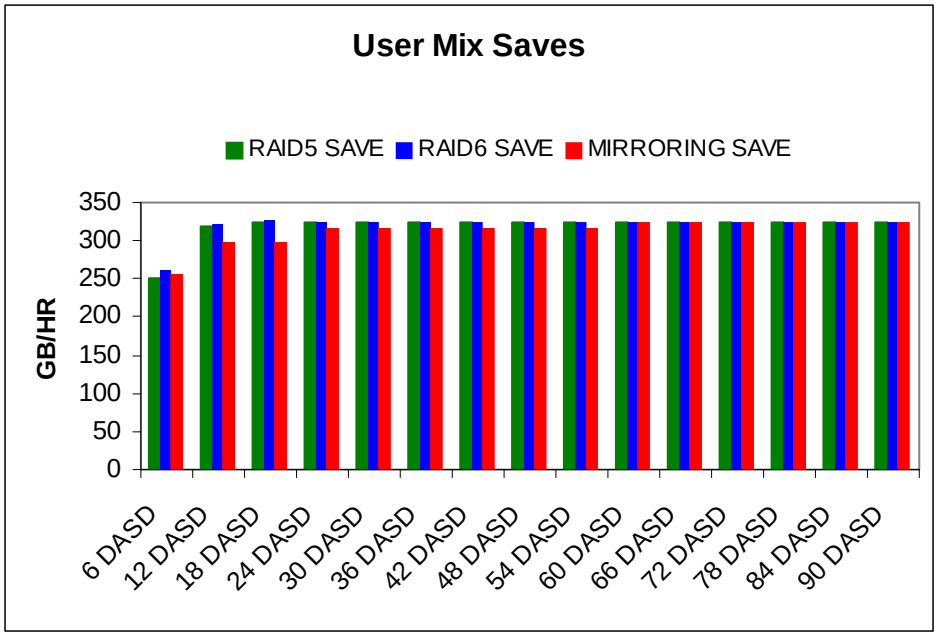


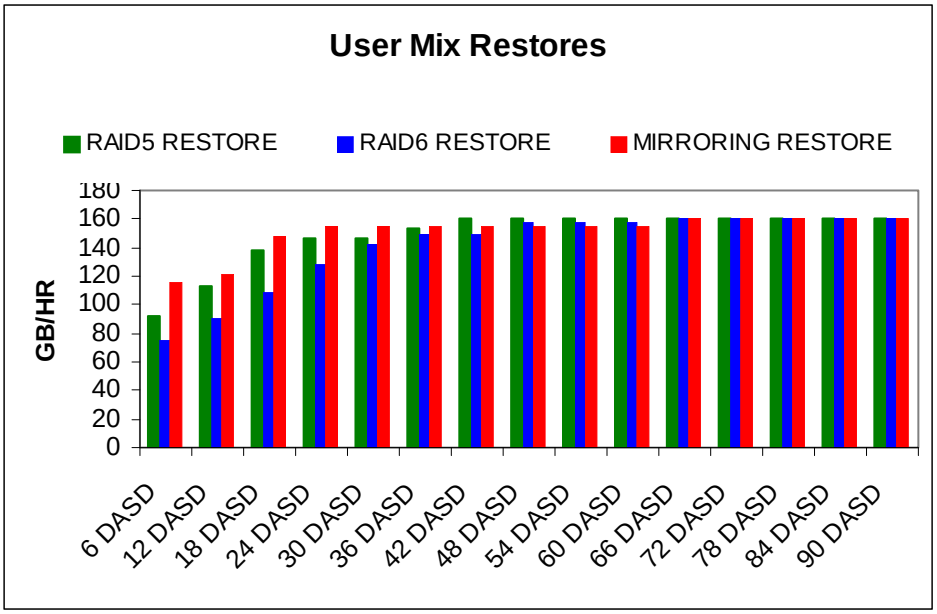
9.16 Save and Restore Scaling using 571E IOAs and U320 15K DASD units to a 3580 Ultrium 3 Tape Drive

A 570 8 way System i was used for the following tests. A user ASP was created with the number of DASD listed in each test . The workload data was then saved to the tape drive , deleted from the system and restored to the user ASP. These charts are very specific to the new IOAs and U320 capable DASD available. For more information on the IOAs and DASD see Chapter 4 of this guide.









9.17 High-End Tape Placement on System i

The current high-end tape drives (ULTRIUM-2 / ULTRIUM-3 and 3592-J1a) need to be placed carefully on the System i buses and HSLs in order to avoid bottlenecking. The following rules-of thumb will help optimize performance in a large-file save environment, and help position the customer for future growth in tape activity:

- Limit the number of drives per fibre tape adapter as follows:
  - For ULTRIUM-2, 3592-J1a, and slower drives, two drives can share a fc 5704 or fc 5761 fibre tape adapter. If running on a 2 GByte loop, a 3rd drive can share a fc 5761 fibre tape adapter
  - For ULTRIUM-3 and TS1120 (3592-E05) drives, each drive should be on a separate fibre tape adapter

- Place the fc 5704 or fc 5761 in a 64-bit slot on a "fast bus" as follows:
  - PCI-X
    - In a 5094/5294 tower use slot C08 or C09.
    - In a 5088/0588 tower use slot C08 or C09. You may need to purchase RPQ #847204 to allow the tower to connect with RIO-G performance
    - In an 0595 or 5095 or 5790 expansion unit, use any valid slot
  - PCI
    - In a 5074/5079/5078 tower, use slot C02, C03 or C04
  - Note
    - "Ensure the fc 5761 is supported on your CPU type"

- Put one fc 5704 or fc 5761 per tower initially. On loops running at 2 GByte speeds, a second fc 5704 card can be added according to the locations recommended above if needed.

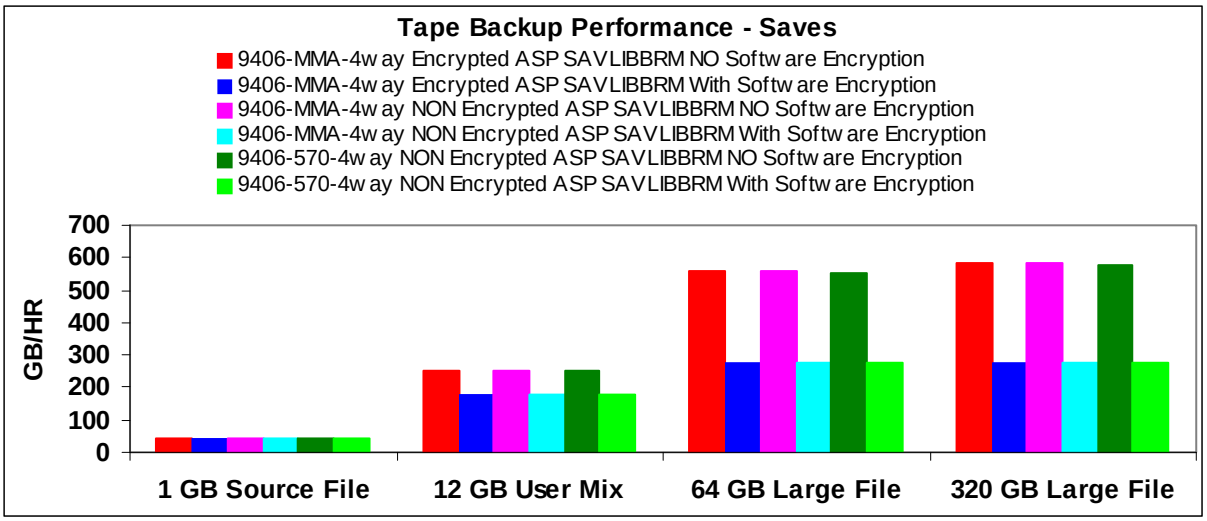
- Spread tape fibre cards across as many HSL's as possible, with maximums as follow
  - On Loops running at 1 GByte (e.g. all loops on 8xx systems, or loops with HSL-1 towers)
    - Maximum of two drives per HSL loop
  - On Loops running at 2 GByte (eg loops with all HSL-2 / RIO-G towers on system i systems)
    - Maximum of six ULTRIUM-2 or 3592-J1a drives per RIO-G loop.
    - Maximum of four ULTRIUM-3 drives or TS1120 (3592-E05) drives per RIO-G loop using the fc 5704 IOA.
    - Maximum of two TS1120 (3592-E05) drives per RIO-G loop using the fc 5761 IOA

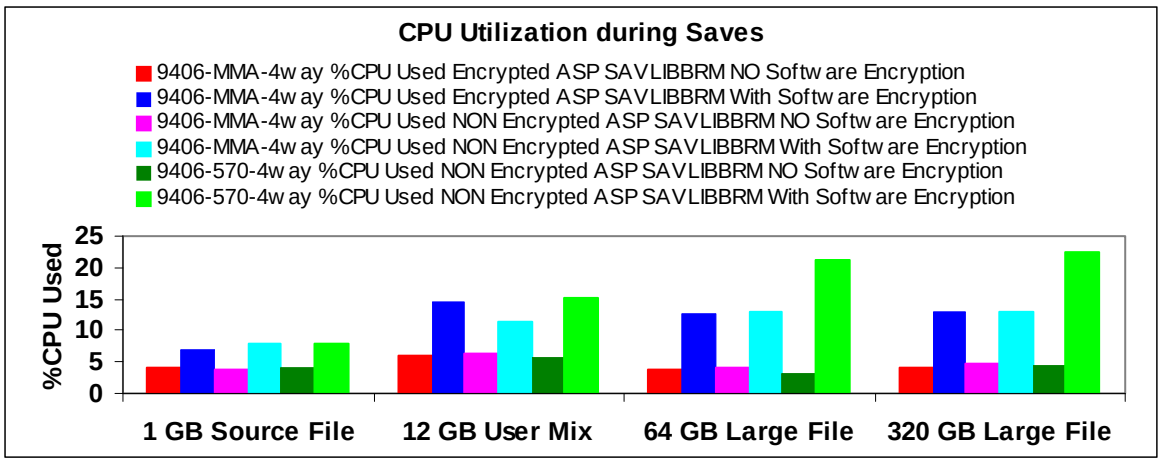
- If Gbit Ethernet cards are present on the system and will be running during the backups, then treat them as though they were ULTRIUM-3 or TS1120 (3592-E05) tape drives when designing the card and HSL placement using the rules above since they can command similar bandwidth

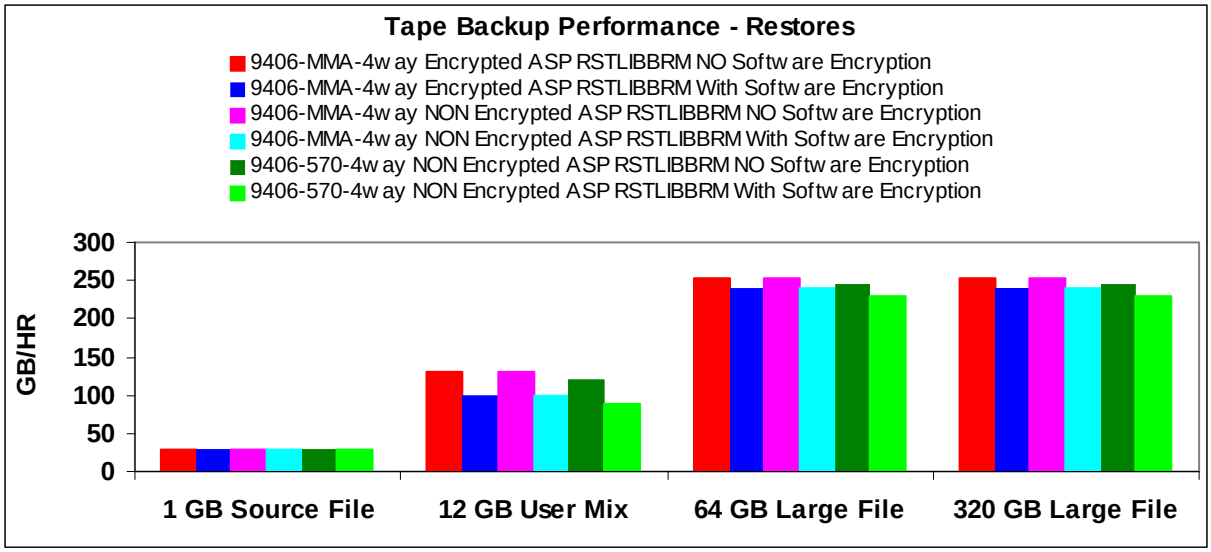
The rules above assume that the customer is running a large-file workload and that all tape drives are active simultaneously. If your customer is running a user-mix tape workload or the high load cards are not running simultaneously, then it may be possible to put more gear on the bus/HSL than shown. There may also be certain card layouts that will allow more drives per bus/tower/HSL, but these need to be reviewed individually.

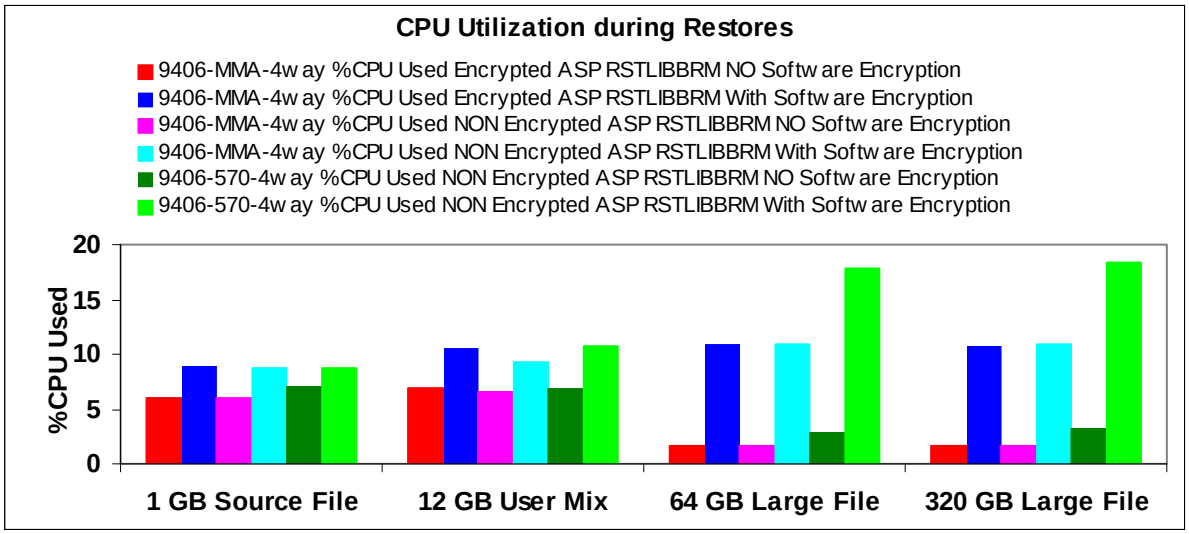
9.18 BRMS-Based Save/Restore Software Encryption and DASD-Based ASP Encryption

The Ultrium-3 was used in the following experiments, which attempt to characterize the effects of BRMS-based save /restore software encryption and DASD-based ASP encryption. Some of the newer tape drives offer hardware encryption as an option but for those who are not looking to upgrade or invest in these tape units at this time, software encryption can be a fair solution. In the experiments we used full processors that were dedicated to the partition. We used a 9406-MMA 4 way partition and a 9406-570 4 way partition. Both systems had 40 GB of main memory. The workload data was located on a single user ASP with 36 - 70GB 15K RPM DASD attached through a 571F IOA. The experiments were not set up to show the best possible environment or to take into account all of the possible hardware environments, instead these experiments were an attempt to portray some of the differences customers might observe if they choose software encryption as a back up strategy over their current non-encrypted environment. Software encryption has a significant impact on save times but only a minor impact to restore times









Performance will be limited to the native drive rates (shown in table 9.1.1) because encrypted data blocks have a very low compaction ratio.

9.19 Tape Device Rates

Note: Measurements for the high speed devices were completed on a 570 4 way system with 2844 IOPs and 2780 IOA's and 180 15K RPM RAID5 DASD units. The smaller tape device tests were completed on a 520 2 way with 75 DASD units. The Virtual tape and *SAVF runs were completed on a 570 ML16 with 256GB of memory and 924 DASD units. The goal of each of the tests is to show the capabilities of the device and so a system large enough to achieve the maximum throughput for that device was used. Customer performance will be dependent on over all systems resources and if those resources match the maximum capabilities of the device. See other sections in this guide about memory, CPU and DASD.

Table 9.19.1
Measurements in (GB/HR) Workload data Saved and Restored from User ASP 2.

Workload S = Save R = Restore		SLR60	SLR100	VXA-2	6279 VXA-320		5755 ½ High ULTRIUM 2	3580 Ultrium 2 5704 2GB Fiber Adapter	3592-J1a 5704 2GB Fiber Adapter	3580 Ultrium 3 5704 2GB Fiber Adapter	3592-E05 Fiber 5704 2GB Fiber Adapter	3592-E05 Fiber fc 5761 4GB Fiber Adapter	*SAVF	Virtual Tape Drive
Release Measurements were done		iV5R4	iV5R4	iV5R3	iV5R4		iV5R4	iV5R3	iV5R3	iV5R4	iV5R4	iV5R4	iV5R3	iV5R4
Source File 1GB	S	17	17	14	19		21	17	17	22	30	30	35	35
	R	19	17	19	19		24	20	20	29	30	30	20	20
User Mix 3GB	S	30	31	33	48			113	130					
	R	30	31	33	48			50	115					
User Mix 12GB	S	32	35	40	70		145	150	180	200	210	210	220	220
	R	30	31	35	53		96	80	120	150	180	180	125	180
Large File 4GB	S	32	34	37				280	280					
	R	32	34	37				280	340					
Large File 32GB	S			41	82		225	350	365	500	560	800		
	R			40	68		175	330	390	500	560	800		
Large File 64GB	S				82		225	350	365	500	560	830	1330	1450
	R				68		175	330	390	500	560	830	1340	1500
Large File 320GB	S									525	580	890	1420	1700
	R									510	570	830	1340	1530
1 Directory Many Objects	S	23	25	27	40		35	65	65	65	65	65	65	70
	R	12	13	13	30		47	14	16	50	60	60	50	60
Many Directories Many Objects	S	25	25	30	40		35	50	50	50	50	50	50	55
	R	9	9	9	20		23	9	9	30	30	30	23	30
Domino Mail Files	S	29	29	35	67		125	190	230	410	500	530	1000	1250
	R	29	29	33	55		110	190	230	420	500	560	1000	1200
Network Storage Space	S	34	34	40	70		125	200	230	350	380	500	1100	1100
	R	34	34	40	56		140	200	260	380	380	490	1050	1100

Table 9.19.2 - iV5R4M0 Measurements on an 5XX 1-way system 8 RAID5 protected DASD Units 8 GB memory Measurements in (GB/HR) all 8 DASD in the system ASP .

Workload S = Save R = Restore		6258 4MM tape Drive	SLR60 from table 9.18.1							
Release Measurements were done		iV5R4M0	iV5R4							
Source File 1GB	S	22	17							
	R	15	19							
User Mix 12GB	S	34	30							
	R	30	30							
Large File 32GB	S	39	32							
	R	37	32							
1 Directory Many Objects	S	12	23							
	R	8	12							
Many Directories Many Objects	S	15	25							
	R	7	9							
Domino Mail Files	S	15	29							
	R	15	29							
Network Storage Space	S	19	34							
	R	19	34							

9.20 Tape Device Rates with 571E & 571F Storage IOAs and 4327 (U320) Disk Units

Save/restore rates of 3580 Ultrium 3 (2Gb and 4Gb Fiber Channel) tape devices and of virtual tape devices were measured on a 570 8-way system with 571E and 571F storage adapters and 714 type 4327 70GB (U320) disk units. Customer performance will be dependent on overall system resources and how well those resources match the maximum capabilities of the device. See other sections in this guide about memory, CPU and DASD.

Table 9.20.1
Measurements in (GB/HR)
Workload data Saved and Restored from User ASP 2.

Workload S = Save R = Restore		2780 Storage IOAs 4326 35GB (U160) DASD (Data from table 9.18.1)	571E/571F Storage IOAs 4327 70GB (U320) DASD			
		3580 Ultrium 3 5704 2Gb Fiber Adapter	3580 Ultrium 3 5704 2Gb Fiber Adapter	3580 Ultrium 3 5761 4Gb Fiber Adapter	3580 Ultrium 4 5761 4Gb Fiber Adapter	Virtual Tape Drive
Release Measurements were done		iV5R4	iV5R4	iV5R4	iV5R4	iV5R4
Source File 1GB	S	22	95	110	55	110
	R	29	40	40	26	40
User Mix 12GB	S	200	290	290	295	345
	R	150	175	175	182	195
Large File 64GB	S	500	510	585	650	1380
	R	500	550	785	760	1230
Large File 320GB	S	525	525	635	650	1420
	R	510	550	785	760	1240
1 Directory Many Objects	S	65	80	80	90	80
	R	50	60	60	45	65
Many Directories Many Objects	S	50	55	60	65	65
	R	30	30	30	25	30
Domino Mail Files	S	410	440	450	550	1410
	R	420	460	460	600	1190
Network Storage Space	S	350	355	410	425	1300
	R	380	405	460	525	1230

9.21 DVD RAM and Optical Library

Table 9.21.1 - iV5R3 Measurements on an 520 2-way system 53 RAID protected DASD Units 16 GB memory Measurements in (GB/HR) ASP 1 (System ASP 23 DASD) ASP 2 (30 DASD) Workload data Saved and Restored from User ASP 2.

Workload S = Save R = Restore		6331 DTACPR *NO	6331 DTACPR *YES	6333 DTACPR *NO	6333 DTACPR *YES	6330 DTACPR *NO	6330 DTACPR *YES		399F Model 200 Optical Library UDO	399F Model 200 Optical Library 14x
Release Measurements were done		V5R3	V5R3	V5R3	V5R3	V5R3	V5R3		V5R3	V5R3
Source File 1GB	S	1.8	9.0	2.2	12.0	3.0	14.0		6	5.3
	R	9.2	21.0	9.8	21.0	9.0	21.0		4.5	4.5
User Mix 3GB	S	1.8	6.0	2.0	7.5	2.6	9.0		6	5.3
	R	9.5	29.0	9.5	29.0	9.5	29.0		14	11.5
Large File 4GB	S	1.8	6.0	2.0	7.2	2.7	9.0		6	5.6
	R	9.7	31.0	9.7	31.0	9.7	31.0		21	16.5
1 Directory Many Objects	S	1.8	1.8	2.2	2.2	2.6	2.6			
	R	7.5	7.5	7.7	7.7	7.8	7.7			
Many Directories Many Objects	S	1.8	1.8	2.2	2.2	2.6	2.6			
	R	5.4	5.4	6.0	6.0	6.0	6.0			
Domino Mail Files	S	1.8	1.8	2.0	2.0	2.6	2.6			
	R	9.6	9.6	9.8	9.8	9.8	9.8			
Network Storage Space	S	1.8	1.8	2.0	2.0	2.6	2.6			
	R	9.6	9.6	9.8	9.8	9.8	9.8			

9.22 Software Compression

The rates a customer will achieve will depend upon the system resources available. This test was run in a very favorable environment to try to achieve the maximum rates. Software compression rates were gathered using the QSRSAVO API. The CPU used in all compression schemes was near 100%. The compression algorithm cannot span CPUs so the fact that measurements were performed on a 24-way system doesn't affect the software compression scenario.

Table 9.22.1 - Measurements on an 840 24-way system 1080 RAID protected DASD Units (GB/HR) 128 GB mainstore

		NSRC1GB	NUMX12GB	SR16GB	Software Compression Ratio
iV5R1	Save	19	135	170	
	Restore	7	45	170	
iV5R2	Save	19	200	480	
	Restore	7	50	480	
iV5R2 Using API DTACPR *LOW	Save		88	108	1.5:1
	Restore		37	57	
iV5R2 Using API DTACPR *MED	Save		26	27	2.7:1
	Restore		23	31	
iV5R2 Using API DTACPR *HIGH	Save		6	6	3:1
	Restore		39	65	

9.23 9406-MMA DVD RAM

Table 9.23.1 - All measurements on an 9406-MMA 4-way system 6 Mirrored DASD in the CEC and 24 RAID5 protected DASD Units attached 32 GB memory Measurements in (GB/HR) all 30 DASD in the system ASP.

Workload S = Save R = Restore		SAS 6331 DTACPR *NO 5X Media	SAS 6331 DTACPR *YES 5X Media	SATA 6331 DTACPR *NO 5X Media	SATA 6331 DTACPR *YES 5X Media				
Release Measurements were done		iV5R4M5	iV5R4M5	iV6R1M0	iV6R1M0				
Source File 1GB	S	3.0	13.4	3.0	11.5				
	R	7.3	9.3	15.0	31.0				
User Mix 3GB	S	2.3	8.0	2.6	8.1				
	R	12.5	28.0	19.5	47.0				
Large File 4GB	S	2.2	8.0	2.6	8.1				
	R	14.0	45.0	20.5	56.0				
1 Directory Many Objects	S	2.3	2.3	2.2	2.2				
	R	9.0	9.0	9.1	9.1				
Many Directories Many Objects	S	2.2	2.2	2.1	2.1				
	R	5.5	5.5	6.4	6.1				
Domino Mail Files	S	2.3	2.3	2.7	2.7				
	R	14.5	14.5	21.0	21.0				

9.24 9406-MMA 576B IOPLess IOA

Table 9.24.1 - iV6R1M0 Measurements on an 9406-MMA 4-way system 200 RAID5 protected DASD Units in the system ASP, attached via 571F IOAs 40 GB memory Measurements in (GB/HR). Two different Virtual tape experiments with 60 RAID5 DASD ASP2 and 120 RAID5 DASD in ASP2

Workload S = Save R = Restore		3580 Ultrium 3 576B 2 Port 4Gb Fiber Adapter	3580 Ultrium 4 576B 2 Port 4Gb Fiber Adapter	3592-E05 Fiber 576B 2 Port 4Gb Fiber Adapter	5746 Half High Ultrium 4 572A Adapter	3592-E06 Fiber 576B 2 Port 4Gb Fiber Adapter	Virtual Tape 60 DASD in ASP2	Virtual Tape 120 DASD in ASP2	Two High Speed Tape Drives on a Single 576B IOA using both ports concurrently
IBM i Release		V6R1M0	V6R1M0	V6R1M0	V6R1M0	V6R1M0	V6R1M0	V6R1M0	V6R1M0
Source File 1GB	S	40	32	34	34	34	40	40	
	R	45	50	50	50	50	42	42	
User Mix 12GB	S	280	234	230	230	230	220	280	
	R	190	210	230	210	230	220	220	
Large File 64GB	S	615	859	885	700	1050	350	770	
	R	590	837	810	700	1000	750	770	1st Drive 2nd Drive
Large File 320GB	S	625	890	920	700	1100	350	770	920 885
	R	590	890	845	700	1000	750	770	485 475
1 Directory Many Objects	S	50	55	55	55	55	50	50	
	R	50	50	50	50	50	50	50	
Many Directories Many Objects	S	40	40	40	40	40	38	38	
	R	26	28	27	27	27	26	26	
Domino Mail Files	S	450	575	580	550	650	330	700	
	R	450	650	650	650	750	700	700	

9.25 BladeCenter H SAS attached LTO4

Table 9.25.1 - In the following table I place the information gathered on the BladeCenter H and BladeCenter S with the SAS attached LTO4 to compare it to the current data collected from an LTO4 on a 9406-MMA using internal DASD. Not a direct comparison but it shows the saves have plenty of resource but the restores are bound by the DASD configuration of the BladeCenter. See previous sections in this chapter to better understand how processors., memory and DASD recources affect Save and Restore operations.

BladeCenter Hardware:Release iV6R1MX Measurements on an JS23 Double Wide 8 way Blade in a BladeCenter H model DS4800 Storage attached via fiber channel connections. System ASP consisted of 12 DDMs in 2 LUNs each LUN consisted of 6 RAID1 DDMs. DATA ASP consisted of 72 DDMs in 12 LUNs each LUN consisted of 6 DDMs configured using RAID1 arrays.

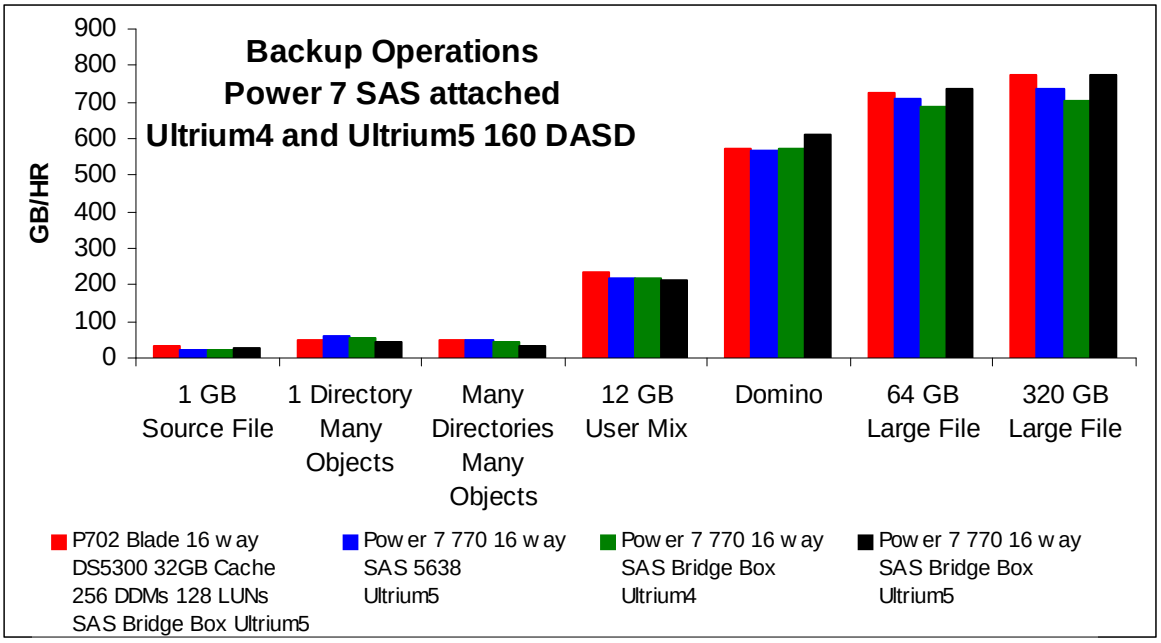
BladeCenter S Hardware: Release V6R1M1 Measurements on a JS12 2 way balde with .2 cpu given to VIOS and 1.8 processors assigned to the IBM i partition.. BladeCenter S internal DASD 8 LUNS all in ASP 1.

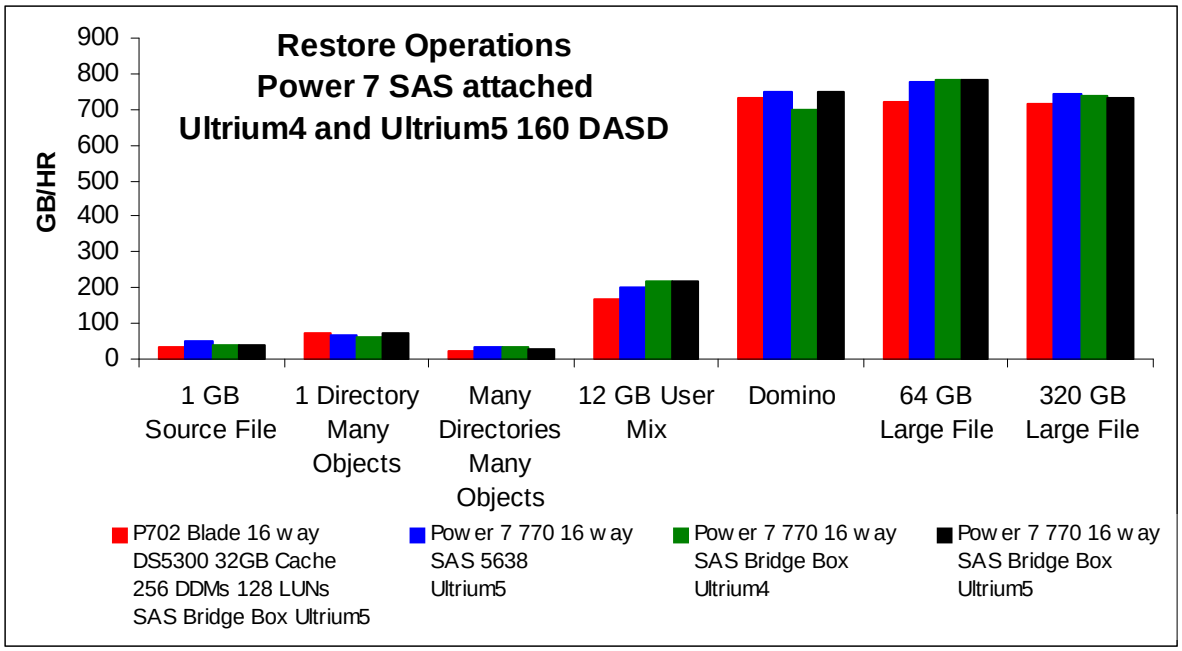
9406 -MMA hardware: iV6R1M0 Measurements on an 9406-MMA 4-way system 200 RAID5 protected DASD Units in the system ASP, attached via 571F IOAs 40 GB memory Measurements in (GB/HR).

Workload S = Save R = Restore		MMA 3580 Ultrium 4 576B 2 Port 4Gb Fiber Adapter	MMA 5746 Half High Ultrium 4 572A Adapter	3580 Ultrium 4 SAS Attached **BladeCenter H**	3580 Ultrium 4 Fiber Channel Attached **BladeCenter H**		3580 Ultrium 4 SAS Attached **BladeCenter S** 6 RAID1 DASD	3580 Ultrium 4 SAS Attached **BladeCenter S** 12 RAID1 DASD
IBM i Release		V6R1M0	V6R1M0	V6R1M0	V6R1M1		V6R1M1	V6R1M1
Source File 1GB	S	32	34	50	34		11	25
	R	50	50	15	30		7	10
User Mix 12GB	S	234	230	300	300		110	150
	R	210	210	90	155		50	55
Large File 64GB	S	859	700	685	830		320	370
	R	837	700	525	800		290	390
Large File 320GB	S	890	700	700	840			
	R	890	700	525	630			
1 Directory Many Objects	S	55	55	85	85		30	30
	R	50	50	30	40		10	15
Many Directories Many Objects	S	40	40	50	50		30	30
	R	28	27	13	17		7	9
Domino Mail Files	S	575	550	550	600		280	320
	R	650	650	520	600		250	300

9.26 SAS Attach Ultrium 4 and Ultrium 5

With SAS BridgeBox and the enclosed Ultrium half height tape drives the IBM lab experiments showed no real difference between the Ultrium 4 and Ultrium 5 devices.

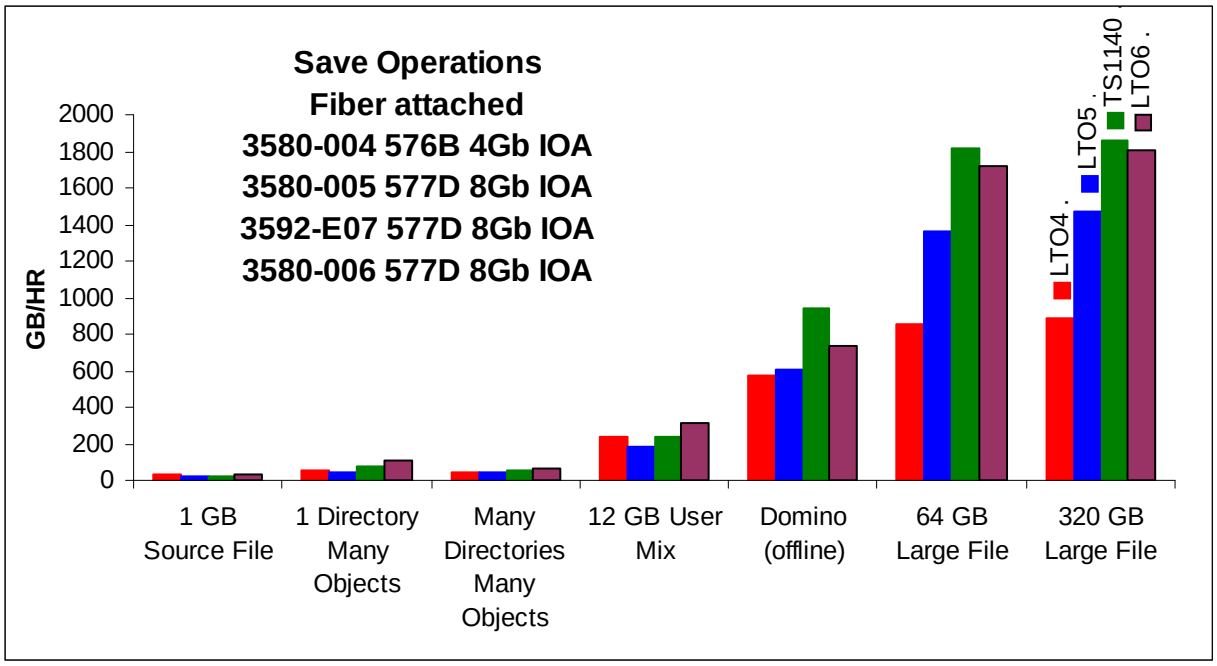


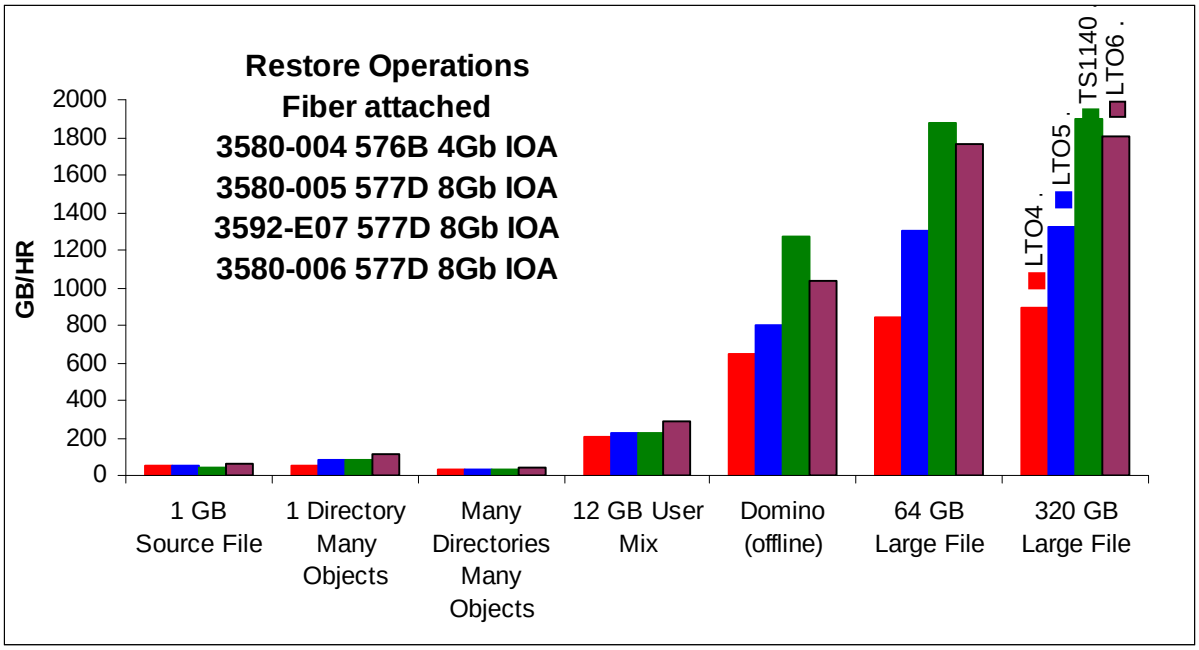


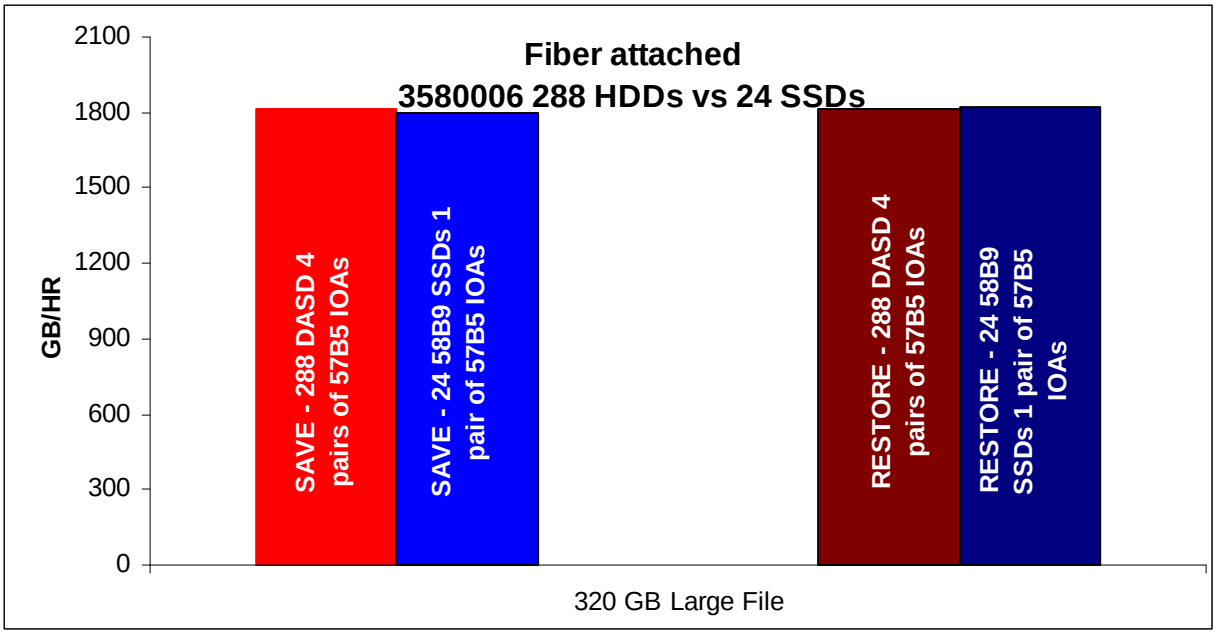
SAVE	SAS Bridge Box Ultrium5	SAS 5638 Ultrium5	SAS Bridge Box Ultrium4	SAS Bridge Box Ultrium5
1 GB Source File	30	23	25.5	25
1 Directory Many Objects	47	53	87.5	41.5
Many Directories Many Objects	47	46	59	34
12 GB User Mix	254.5	242	296.5	209
Domino	628.5	556.5	578.5	608
64 GB Large File	734	710	686.5	720
320 GB Large File	774.5	736.5	705	795
RESTORE				
1 GB Source File	32.5	47	42.5	41
1 Directory Many Objects	70.5	67	64.5	72.5
Many Directories Many Objects	25.5	31	31.5	28
12 GB User Mix	160	198.5	235	214.5
Domino	728.5	760	696	766
64 GB Large File	727	775	802.5	781.5
320 GB Large File	715	743.5	766.5	733

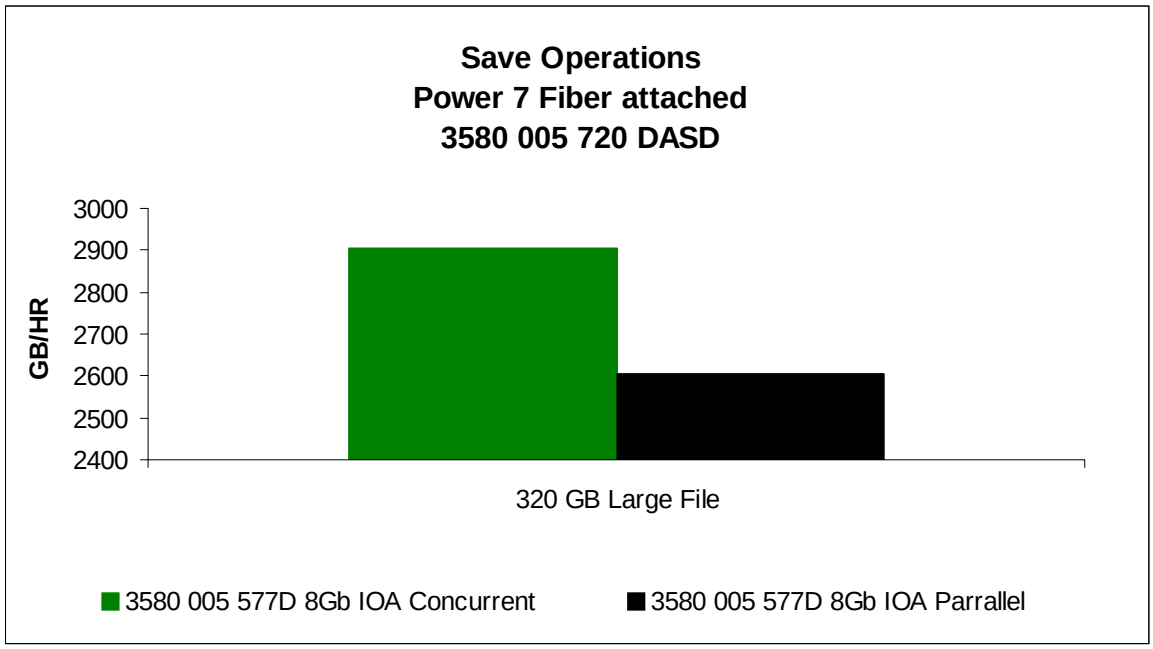
9.27 Fiber attach 3580-004, 3580-005, 3592-E07 and 3580-006

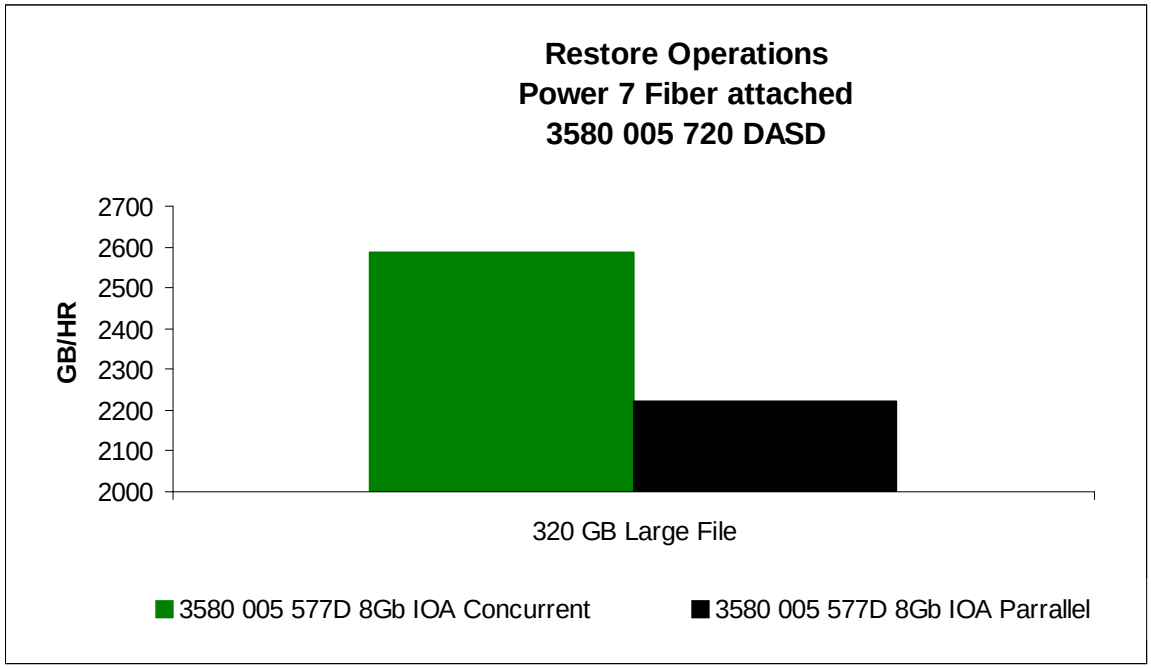
	LTO4	LTO5	TS1140	LTO6
	9117-MMA 16 Way With 200 DASD	9179-MHC With 720 DASD On 574E IOAs EXP 12 3Gb SAS Drawers	9179-MHC With 720 DASD on 574E IOAs EXP 12 3Gb SAS Drawers	9179-MHD With 288 DASD on 57B5 IOAs 5887 6Gb SAS Drawers
	3580 004 576B 4Gb IOA	3580 005 577D 8Gb IOA	3592 E07 577D 8Gb IOA	3580 006 577D 8Gb IOA
SAVE	GB/hr	GB/hr	GB/hr	GB/hr
1 GB Source File	32	25	26	28
1 Directory Many Objects	55	46.5	77	110
Many Directories Many Objects	40	44.5	56	67
12 GB User Mix	234	220	241	312
Domino (offline)	575	605	937	740
64 GB Large File	859	1366	1814	1720
320 GB Large File	890	1475	1861	1810
RESTORE	GB/hr	GB/hr	GB/hr	GB/hr
1 GB Source File	50	48	41	59
1 Directory Many Objects	50	78.5	78	112
Many Directories Many Objects	28	33.5	29	42
12 GB User Mix	210	225	227	286
Domino (offline)	650	803	1273	1038
64 GB Large File	837	1307.5	1873	1760
320 GB Large File	890	1327	1895	1810





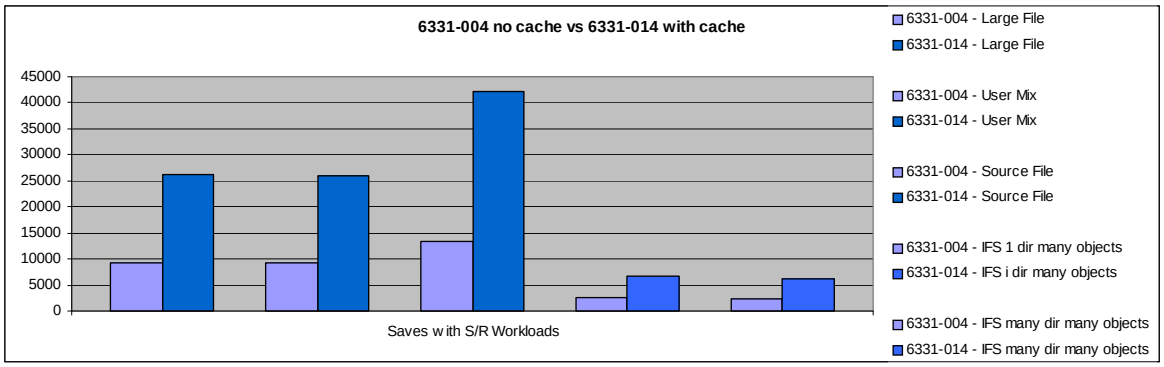


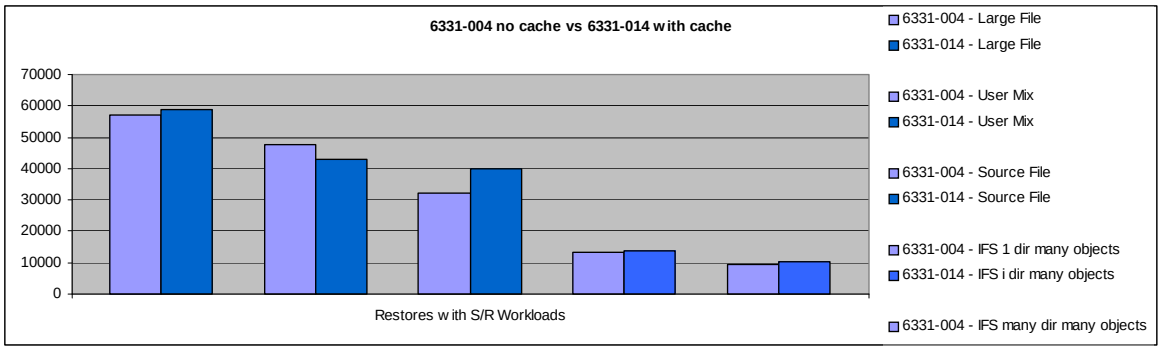




9.28 6331-014 DVD Performance

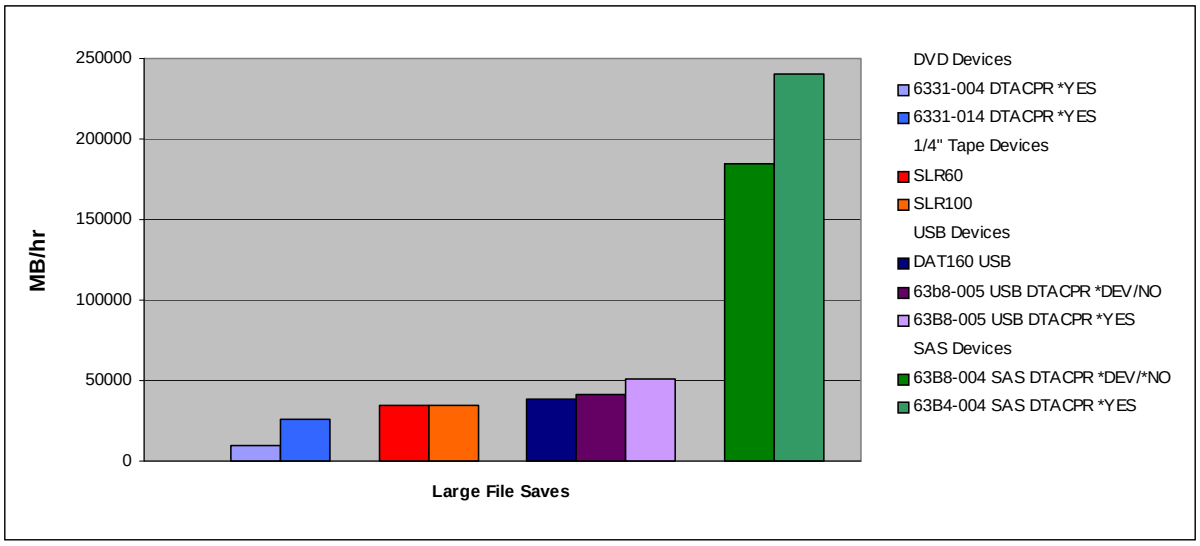
		6331-004	6331-014
		MB/hr	MB/hr
Large File Workload	SAV	9170	26222
	RST	57146	58788
User Mix	SAV	9235	25884
	RST	47500	43106
Source File	SAV	13250	42195
	RST	32216	40068
1 Directory Many Objects	SAV	2540	6674
	RST	13220	13910
Many Directories Many Objects	SAV	2430	6105
	RST	9586	10458

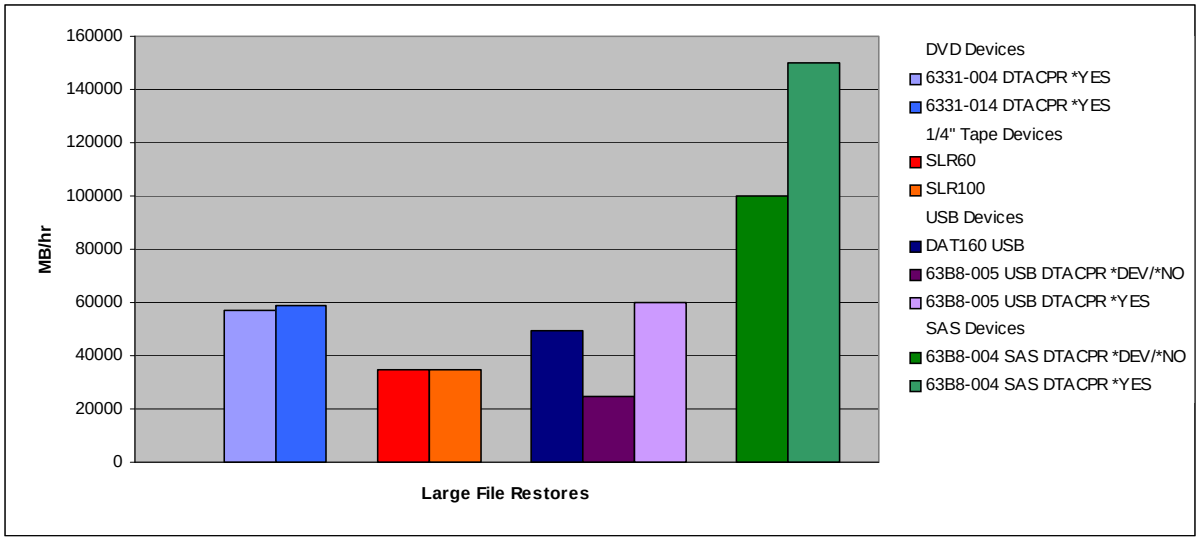


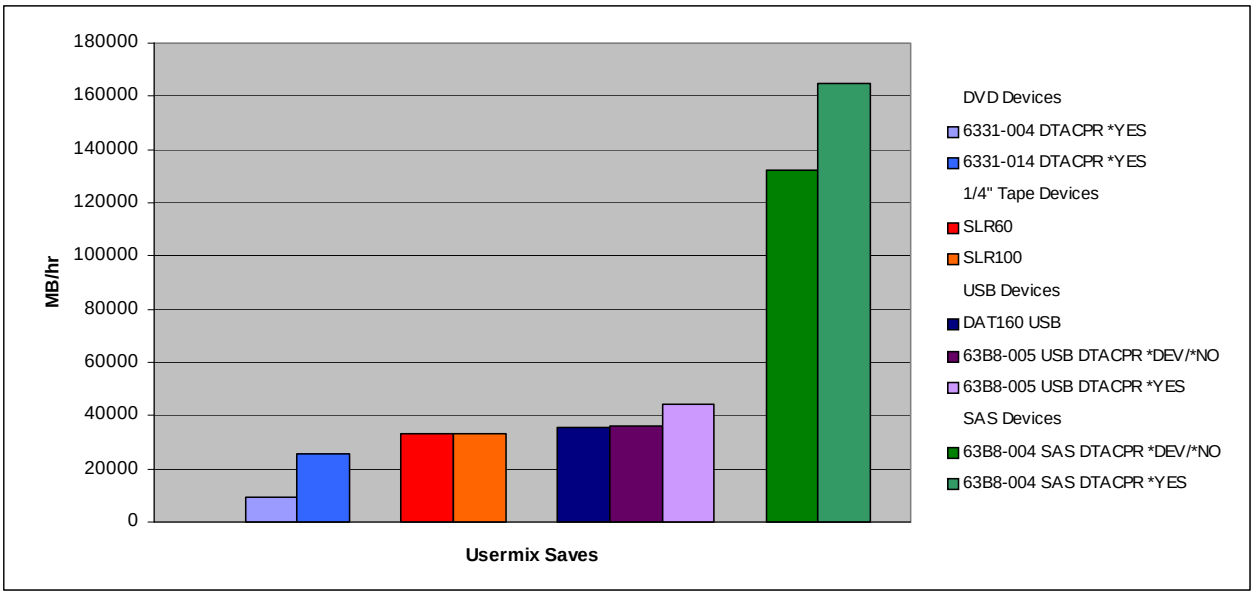


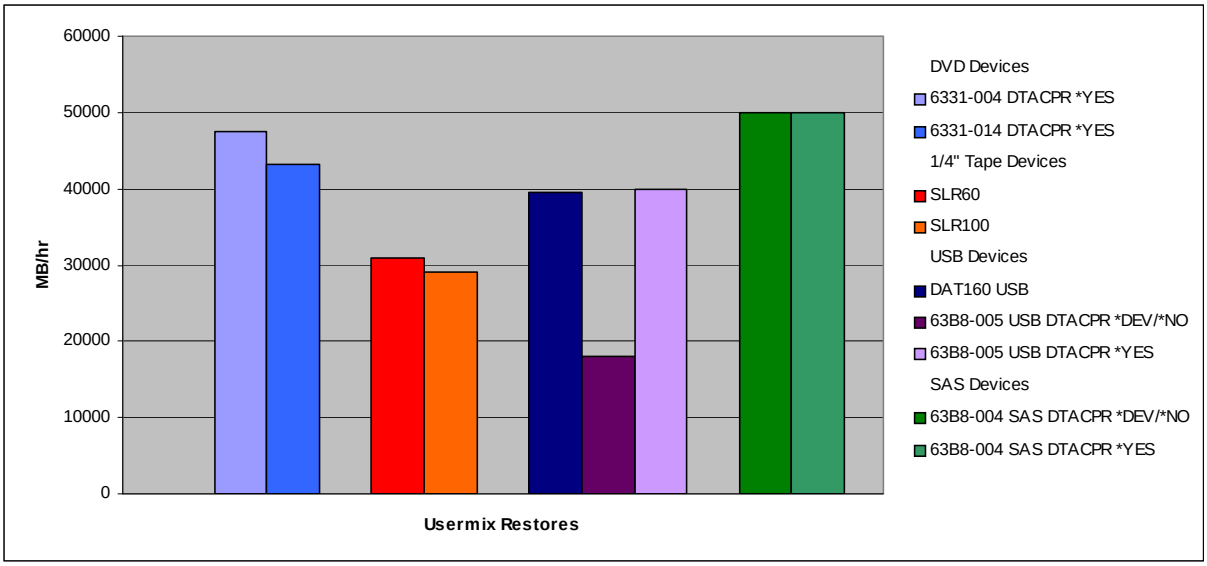
9.29 RDX Device Performance

Workload	Operation	6331-004 DTACPR *Yes	6331-014 DTACPR *Yes	SLR60	SLR100	DAT160 USB	63B8 005 USB DTACPR *DEV/*NO	63B8 005 USB DTACPR *YES	63B8 004 SAS DTACPR *DEV/*NO	63B8 004 SAS DTACPR *YES
		MB/hr	MB/hr	MB/hr	MB/hr	MB/hr	MB/hr	MB/hr	MB/hr	MB/hr
Large File	SAV	9170	26222	34500	35000	38000	41000	51000	185000	240000
	RST	57146	58788	34500	35000	49500	25000	60000	100000	150000
User Mix	SAV	9235	25884	33000	33000	35500	36000	44000	132000	165000
	RST	47500	43106	31000	29000	39500	18000	40000	50000	50000
Source File	SAV	13250	42195	17000	18000	43000	27000	32000	46000	50000
	RST	32216	40068	19000	18000	32500	13000	17000	16000	16000
1 Directory Many Objects	SAV	2540	6674	18000	18000	21000	14000	14000	90000	90000
	RST	13220	13910	19000	19000	20000	14000	14000	19000	19000
Many Directories Many Objects	SAV	2430	6105	26000	26000	17000	9000	9000	56000	56000
	RST	9586	10458	15000	15000	14000	9000	9000	10000	10000









Chapter 10. Batch Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

In a commercial environment, batch workloads tend to be I/O intensive rather than CPU intensive. The factors that affect batch throughput for a given batch application include the following:

- Memory (Pool size)
- CPU (processor speed)
- DASD (number and type)
- System tuning parameters

10.1 Effect of CPU Speed on Batch

The capacity available from the CPU affects the run time of batch applications. More capacity can be provided by either a CPU with a higher CPW value, or by having other contending applications on the same system consuming less CPU.

Conclusions/Recommendations

- For CPU-intensive batch applications, run time may scale inversely with a faster CPU. This assumes that the number synchronous disk I/Os are only a small factor. Do not use CPW ratings to compare batch performance -- especially between processor families.

- For I/O-intensive batch applications, run time may not decrease with a faster CPU. This is because I/O subsystem time would make up the majority of the total run time.

- Batch workload windows may be reduced by overlapping or multithreading processing functions. The IBM i operating system can provide a level of multithreading for I/O and database operations.

- Do not use CPW ratings to predict the batch window times when comparing from one processor family to another (e.g. POWER5 to POWER6 or POWER6 to POWER7). CPW ratings are provided as an indicator of OLTP performance and should not be used for other workload types.

10.2 Effect of DASD Type on Batch

For batch applications that are I/O-intensive, the overall batch performance is very dependent on the speed of the I/O subsystem. Depending on the application characteristics, batch performance (run time) will be improved by having DASD that has:

- faster average service times
- read ahead buffers
- write caches

Consider using SSD drives for workloads that execute a large number of read operations.

Additional information on DASD devices in a batch environment can be found in Chapter 4, “Internal Storage Performance”

10.3 Tuning Parameters for Batch

There are several system parameters that affect batch performance. The magnitude of the effect for each of them depends on the specific application and overall system characteristics. Some general information is provided here.

- **Expert Cache**

  Expert Cache does not start to provide improvement unless the following are true for a given workload. These include:
  - the application that is running is disk intensive, and disk I/O's are limiting the throughput.
  - the processor is under-utilized, at less than 60%.
  - the system must have sufficient main storage.

  For Expert Cache to operate effectively, there must be spare CPU, so that when the average disk access time is reduced by caching in main storage, the CPU can process more work.

  To set Expert Cache, you can use the Change Shared Storage Pool (CHGSHRPOOL) CL command. Simply set the PAGING option to *CALC. For advanced users, the Change Pool Tuning Information (QWCCHGTN) API is available to tune storage pools.

- **Job Priority**

  Batch jobs can be given a priority value that will affect how much CPU processing time the job will get. For a system with high CPU utilization and a batch job with a low job priority, the batch throughput may be severely limited. Likewise, if the batch job has a high priority, the batch throughput may be high at the expense of interactive job performance.

- **Dynamic Priority Scheduling**

  See 19.2, “Dynamic Priority Scheduling” for details.

- **Application Techniques**

  The batch application can also be tuned for optimized performance. Some suggestions include:

  - Breaking the application into pieces and having multiple batch threads (jobs) operate concurrently. Since batch jobs are typically serialized by I/O, this will decrease the overall required batch window requirements.

  - Reduce the number of opens/closes, I/Os, etc. where possible.

  - If you have a considerable amount of main storage available, consider using the Set Object Access (SETOBJACC) command. This command pre-loads the complete database file, database index, or program into the assigned main storage pool if sufficient storage is available . The objective is to improve performance by eliminating disk I/O operations.

  - If communications lines are involved in the batch application, try to limit the number of communications I/Os by doing fewer (and perhaps larger) larger application sends and receives. Consider blocking data in the application. Try to place the application on the same system as the frequently accessed data.

- If you are using SQL functions, consider tuning your environment to reduce response time. You may be able to reduce I/O and pathlength by creating efficient indexes. You may also be able to make use of the hardware multithreading functions such as SMT by utilizing the DB2 SMP (symmetric multiprocessing) feature. For more information, see section 4.6.

10.4 System Sizing for Batch workloads

Single-threaded workloads tend to have completely different system characteristics from the CPW rating. As a result, do not use the CPW rating to estimate system capacity requirements for this class of workloads.

Chapter 11. PowerHA SystemMirror Performance

For the latest information on PowerHA SystemMirror performance, please refer to the performance section of the PowerHA SystemMirror wiki at www.ibm.com/developerworks/ibmi/ha.

Chapter 12. DB2 for i Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

This chapter provides a summary of the new performance features of DB2 for i on 7.1 along with a section of performance references for DB2 for i.

12.1 New for DB2 for i on 7.1

In DB2 for i on 7.1 there are several performance enhancements as follows:

- Adaptive Query Processing along with Global Statistics
- SQE support for Select/Omit logical files and Sparse indexes
- Encoded Vector Indexes with included aggregation
- Concurrency improvement using the new concurrent access resolution attributes, USE CURRENTLY COMMITTED and WAIT FOR OUTCOME

Adaptive Query Processing is new learning optimizer technology in DB2 for i. With the increasing complexity of queries and volumes of data processed, the consequence of miscalculating complex plans can be performance problems. The SQL Query Engine (SQE) uses a technique called Adaptive Query Processing (AQP) to improve query plans by analyzing actual query run time statistics and feeding back that learned knowledge in to subsequent query optimizations. AQP improves query plans of currently executing longer running queries and future runs of queries. To provide this capability, there are three main parts to AQP.

1.) Global Statistics Cache (GSC): The GSC is the " Statistics Cache" repository of statistical information gathered from actual query runs. When SQE observes a large discrepancy between estimated record counts and actual observed values, an entry is made in the GSC. This entry provides the optimizer with more accurate statistical information for subsequent optimizations of the currently running query and other queries for which the optimizer requests the same statistic.
2.) AQP Request Task Support: This support runs after a query completes. If a quick check at the end of a query indicates the plan did not run according to estimates, the AQP Request Task is signalled to more thoroughly examine the actual run data vs. the optimizer estimated plan information. This processing is done very efficiently in a separate task to minimize the performance effect on user applications. Estimated record counts are compared to the actual values. If significant discrepancies are noted, the AQP Request Task stores the observed statistic in the GSC. The AQP Request Task Support can also make specific recommendations for improving the query plan the next time the query runs.
3.) AQP Handler: The AQP Handler runs in a thread parallel to a running query and observes its progress. The AQP handler wakes up after a query runs for at least 2 seconds without returning any rows. Its job is to analyze the actual statistics from the partial query run, diagnose, and recover from join order problems. The query is reoptimized using partial observed statistics or specific join order recommendations or both. If this optimization results in a new plan, the old plan is terminated and the query is restarted with the new plan, provided the query has not returned any results.

AQP is very powerful technology which can provide dramatic performance improvements for individual queries. The technology has been effectively designed to minimize overhead and to improve plans whose estimates have large discrepancies.

In DB2 for i on 7.1 SQE supports the ability to cost and use Select/Omit (S/O) logical files and Sparse Indexes. The optimizer now has the ability to compare/match DDS record selection in the S/O logical file

to the Where selection specified in the SQL statement. If the DDS record selection is a superset of the Where selection for the SQL statement, then the S/O logical file can be used to implement the query. The optimizer will perform estimates over the selection predicates built into the S/O logical file. SQE now offers the same level of support for S/O logical files as that of CQE (Classic Query Engine).

Sparse indexes, indexes created using WHERE selection predicates, are also now supported by SQE. The SQE optimizer has the ability to compare/match the Where selection in the Sparse index to the Where selection specified in the SQL statement. If the Where selection in the Sparse index is a superset of the Where selection for the SQL statement, then the Sparse index can be used to implement the query. The optimizer can perform estimates over the selection built into the Sparse index. Sparse index consideration applies to both radix and encoded vector indexes.

Support for Select/Omit logical files and Sparse Indexes in SQE offers more tuning options to improve performance with DB2 for i on 7.1. The Select/Omit logical file support provides compatibility so that logical files created for native database access may now also be used by SQE. The new sparse index support provides another option to indexing strategy to improve query performance.

Also new for index support in DB2 for i is the addition of ready-made aggregates using Encoded Vector Indexes (EVI). These aggregates are specified using the INCLUDE keyword on the CREATE ENCODED VECTOR INDEX request. The INCLUDE keyword supports numeric aggregate results, such as SUM, COUNT, AVG, or VARIANCE, over non-key data where the grouping is over the corresponding EVI symbol table defined keys. The aggregate can be over a single column or a derivation. The aggregates are maintained in real time as rows are inserted, updated, or deleted from the corresponding table.

Shown below is an example of how an EVI with an included aggregation could be used for a query.
The EVI created with CREATE ENCODED VECTOR INDEX evit1 ON t1(col1) INCLUDE(AVG(col2))
would match the grouping and aggregations in the query,

```
SELECT AVG(col2)
FROM t1
GROUP BY col1.
```

The new capability of including aggregates with Encoded Vector Indexes is a powerful new option for performance tuning SQL queries by facilitating index only access to ready made aggregate values. They can play a vital role as part of an overall tuning strategy using indexes and MQTs. As EVIs are automatically maintained, they can potentially replace some MQTs, eliminating the need for manual refresh.

DB2 for i on 7.1 has a new feature for improving concurrency, new concurrent access resolution attributes, USE CURRENTLY COMMITTED and WAIT FOR OUTCOME. In DB2 for i 6.1, with isolation level CS (cursor stability), only committed and consistent data can be accessed. One transaction will be suspended when it requests a lock that is already held by another transaction and cannot be shared, causing the suspended transaction to temporarily stop running. In DB2 for i on 6.1, the SKIP LOCKED DATA phrase as a SELECT attribute was introduced, but this feature only allows the transaction to skip the rows being incompatibly locked. DB2 for i on 7.1 has a new feature, USE CURRENTLY COMMITTED, which allows users to access the currently committed (last committed) image of data when lock contention is encountered on the data row. This feature also implements a way to direct the database manager to wait for the outcome when encountering data in the process of being updated (WAIT FOR OUTCOME). This allows the database to manage the serialization for certain types of applications which blocks reading when updates are in progress.

The currently committed semantics will only affect read-only queries running with isolation level CS being processed via SQE. Also, USE CURRENTLY COMMITTED does not guarantee that the currently committed

data will be used when updates are in progress, but allows the query to be processed with currently committed data when possible and appropriate. If the currently committed data cannot be accessed in a relatively fast and efficient manner, the database manager will revert back to wait for the outcome. The concurrent access resolution values of USE CURRENTLY COMMITTED and SKIP LOCKED DATA can be used to improve concurrency by avoiding lock waits. However, care must be used when using these options because they might affect application functionality.

DB2 for i on 7.1 has new powerful capabilities to improve performance including Adaptive Query Processing, SQE support for Select/Omit logical files and Sparse indexes, Encoded Vector Indexes with included aggregation, and new concurrent access resolution attributes. More information on each of these new capabilities can be found in the *Database Performance and Query Optimization* manual.

12.2 Performance References for DB2 for i

1. The home page for DB2 for i is found at http://www-1.ibm.com/servers/eserver/iseries/db2/ This web site includes the recent announcement information, white paper and technical articles, and DB2 education information.

2. The IBM i information center section on *DB2 for i* under *Database and file systems* has information on all aspects of DB2 for i including the section *Monitor and Tune database* under *Administrative topics*. This can be found at http://www.ibm.com/eserver/iseries/infocenter

3. Information on creating efficient running queries and query performance monitoring and tuning is found in the DB2 for i *Database Performance and Query Optimization* manual. This document contains detailed information on access methods, the query optimizer, and optimizing query performance including using database monitor to monitor queries, using QAQQINI file options and using indexes. To access this document look in the Printable PDF section in the IBM i information center.

4. The IBM i redbooks provide detailed performance information on a variety of topics for DB2. The redbook repository is located at http://publib-b.boulder.ibm.com/Redbooks.nsf/portals/systemi.

Chapter 13. JDBC and ODBC Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

DB2 for i can be accessed through many different interfaces. Among these interfaces are: Windows .NET, OLE DB, Windows database APIs, ODBC and JDBC. This chapter will focus on access through JDBC and ODBC by providing programming and tuning hints as well as links to detailed information.

13.1 DB2 for i access with JDBC

Access to the IBM i data from portable Java applications can be achieved with the universal database access APIs available in JDBC (Java Database Connectivity). There are two JDBC drivers for IBM i. The Native JDBC driver is a type 2 driver. It uses the SQL Call Level Interface for database access and is bundled in the IBM i Developer Kit for Java. The JDBC Toolbox driver is a type 4 driver which is bundled in the IBM i Toolbox for Java. In general, the Native driver is chosen when running on the IBM i server directly, while the Toolbox driver is typically chosen when accessing data on the IBM i server from another machine. The Toolbox driver is typically used when accessing IBM i data from a Windows machine, but it could be used when accessing the IBM i server from any Java capable system. More detailed information on which driver to choose may be found in the JDBC references.

JDBC Performance Tuning Tips

JDBC performance depends on many factors ranging from generic best programming practices for databases to specific tuning which optimizes JDBC API performance. Tips for both SQL programming and JDBC tuning techniques to improve performance are included here.

- In general when accessing a database it takes less time to retrieve smaller amounts of data. This is even more significant for remote database access where the data is sent over a network to a client. For good performance, SQL queries should be written to retrieve only the data that is needed. Select only needed fields so that additional data is not unnecessarily retrieved and sent. Use appropriate predicates to minimize row selection on the server side to reduce the amount of data sent for client processing.

- Follow the 'Prepare once, execute many times' rule of thumb. For statements that are executed many times, use the PreparedStatement object to prepare the statement once. Then use this object to do subsequent executes of this statement. This significantly reduces the overhead of parsing and compiling the statement every time it is executed.

- Do not use a PreparedStatement object if an SQL statement is run only one time. Compiling and running a statement at the same time has less overhead than compiling the statement and running it in two separate operations.

- Consider using JDBC stored procedures. Stored procedures can help reduce network communication time and traffic which improves response time. Java supports stored procedures via CallableStatement objects.

- Turn off autocommit, if possible. Explicitly manage commits in the application, but do not leave transactions uncommitted for long periods of time.

- Use the lowest isolation level required by the application. Higher isolation levels can reduce performance levels as more locking and synchronization are required. Transaction levels in order of increasing level are: TRANSACTION_NONE, TRANSACTION_READ_UNCOMMITTED, TRANSACTION_READ_COMMITTED, TRANSACTION_REPEATABLE_READ, TRANSACTION_SERIALIZABLE

- Reuse connections. Minimize the opening and closing of connections where possible. These operations are very expensive. If possible, keep connections open and reuse them. A connection pool can help considerably.

- Consider use of Extended Dynamic support. It generally provides better performance by caching the SQL statements in SQL packages on the IBM i.

- Use appropriate cursor settings. Use a fetch forward only cursor type if the data does not need to be scrollable. Use read only cursors for retrieving data which will not be updated.

- Use block inserts and batch updates.

- Tune connection properties to maximize application performance. The connection properties are explained in the driver documentation. Among the properties are 'block size' and 'data compression' which should be tuned as follows:

    1. Choose the right 'block size" for the application. 'block size' specifies the amount of data to retrieve from the server and cache on the client. For the Toolbox driver 'block size' specifies the transfer size in kilobytes, with 32 as the default. For the native driver 'block size' specifies the number of rows that will be fetched at a time for a result set, with 32 as the default. When larger amounts of data are retrieved a larger block size may help minimize communication time.
    2. The Toolbox driver has a 'data compression' property to enable compressing the data blocks before sending them to the client. This is set to true by default. In general this gives better response time, but may use more CPU.

References for JDBC

- The IBM i Information Center
  Http://publib.boulder.ibm.com/iseries/

- The home page for Java and DB2 for i
  http://www-03.ibm.com/systems/i/software/db2/javadb2.html

13.2 DB2 for i access with ODBC

ODBC (Open Database Connectivity) is a set of API's which provide clients with an open interface to any ODBC supported database. The ODBC APIs are part of IBM i Access.

In general, the JDBC Performance tuning tips also apply to the performance of ODBC applications:

- Employ efficient SQL programming techniques to minimize the amount of data processed

- Prepared statement reuse to minimize parsing and optimization overhead for frequently run queries
- Use stored procedures when appropriate to bundle processing into fewer database requests
- Consider extended dynamic package support for SQL statement and package caching
- Process data in blocks of multiple rows rather than single records when possible (e.g. Block inserts)

In addition for ODBC performance ensure that each statement has a unique statement handle. Sharing statement handles for multiple sequential SQL statements causes DB2 on i to do FULL OPEN operations since the database cursor can not be reused. By ensuring that an SQLAllocStmt is done before any SQLPrepare or SQLExecDirect commands, database processing can be optimized. This is especially important when a set of SQL statements are executed in a loop. Ensuring each SQL statement has its own handle reduces the DB2 overhead.

Tools such as ODBC Trace (available through the ODBC Driver Manager) are useful in understanding what ODBC calls are made and what activity occurs as a result. Client application profilers may also be useful in tuning client applications. These are often included in application development toolkits.

ODBC Performance Settings

You may be able to further improve the performance of your ODBC application by configuring the ODBC data source through the Data Sources (ODBC) administrator in the Control Panel. Listed below are some of the parameters that can be set to better tune the performance of the IBM i Access ODBC Driver. The ODBC performance parameters discussed in detail are:

- Prefetch
- ExtendedDynamic
- RecordBlocking
- BlockSizeKB
- LazyClose
- LibraryView

Prefetch : The Prefetch option is a performance enhancement to allow some or all of the rows of a particular ODBC query to be fetched at PREPARE time. We recommend that this setting be turned ON. However, if the client application uses EXTENDED FETCH (SQLExtendedFetch) this option should be turned OFF.

ExtendedDynamic: Extended dynamic support provides a means to "cache" dynamic SQL statements on the IBM i server. With extended dynamic, information about the SQL statement is saved away in an SQL package object on the IBM i the first time the statement is run. On subsequent uses of the statement, IBM i Access ODBC recognizes that the statement has been run before and can skip a significant part of the processing by using the information saved in the SQL package. Statements which are cached include SELECT, positioned UPDATE and DELETE, INSERT with subselect, DECLARE PROCEDURE, and all other statements which contain parameter markers.

All extended dynamic support is application based. This means that each application can have its own configuration for extended dynamic support. Extended dynamic support as a whole is controlled through the use of the ExtendedDynamic option. If this option in not selected, no packages are used. If the option is selected (default) custom settings per application can be configured with the “Custom Settings Per Application” button. When this button is clicked a “Package information for application” window pops up and package library and name fields can be filled in and usage options can be selected.

Packages may be shared by several clients to reduce the number of packages on the IBM i server. To enable sharing, the default libraries of the clients must be the same and the clients must be running the same application. Extended dynamic support will be deactivated if two clients try to use the same package but have different default libraries. In order to reactivate extended dynamic support, the package should be deleted from the IBM i and the clients should be assigned different libraries in which to store the package(s).

Package Usage: The default and preferred performance setting enables the ODBC driver to use the package specified and adds statements to the package as they are run. If the package does not exist when a statement is being added, the package is created on the server.

Considerations for using package support: It is recommended that if an application has a fixed number of SQL statements in it, a single package be used by all users. An administrator should create the package and run the application to add the statements from the application to the package. Once that is done, configure all users of the package to not add any further statements but to just use the package. Note that for a package to be shared by multiple users each user must have the same default library listed in their ODBC library list. This is set by using the ODBC Administrator.

Multiple users can add to or use a given package at the same time. Keep in mind that as a statement is added to the package, the package is locked. This could cause contention between users and reduce the benefits of using the extended dynamic support.

If the application being used has statements that are generated by the user and are ad hoc in nature, then it is recommended that each user have his own package. Each user can then be configured to add statements to their private package. Either the library name or all but the last 3 characters of the package name can be changed.

RecordBlocking: The RecordBlocking switch allows users to control the conditions under which the driver will retrieve multiple rows (block data) from the IBM i. The default and preferred performance setting to Use Blocking will enable blocking for everything except SELECT statements containing an explicit "FOR UPDATE OF" clause.

BlockSizeKB (choices 2 through 512): The BlockSizeKB parameter allows users to control the number of rows fetched from the IBM i per communications flow (send/receive pair). This value represents the client buffer size in Kilobytes and is divided by the size of one row of data to determine the number of rows to fetch from the IBM i in one request. The primary use of this parameter is to speed up queries that send a lot of data to the client. The default value 32 will perform very well for most queries. If you have the memory available on the client, setting a higher value may improve some queries.

LazyClose: The LazyClose switch allows users to control the way SQLClose commands are handled by the IBM i Access ODBC Driver. The default and preferred performance setting enables Lazy Close. Enabling LazyClose will delay sending an SQLClose command to the IBM i until the next ODBC request is sent. If Lazy Close is disabled, a SQLClose command will cause an immediate explicit flow to the IBM i to perform the close. This option is used to reduce flows to the IBM i, and is purely a performance enhancing option.

LibraryView: The LibraryView switch allows users to control the way the IBM i Access ODBC Driver deals with certain catalog requests that ask for all of the tables on the system. The default and preferred performance setting 'Default Library List' will cause catalog requests to use only the libraries specified in the default library list when going after library information. Setting the LibraryView value to 'All

libraries on the system' will cause all libraries on the system to be used for catalog requests and may cause significant degradation in response times due to the potential volume of libraries to process.

References for ODBC

- *DB2 Universal Database for IBM i SQL Call Level Interface (ODBC)*
  is found under the IBM i Information Center under Printable PDFs and Manuals

- The IBM i Information Center
  Http://publib.boulder.ibm.com/iseries/

- Microsoft ODBC webpage
  http://msdn2.microsoft.com/en-us/library/ms710252.aspx

Chapter 14. Java Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

Highlights:
- Introduction
- What's new in V6R1
- IBM Technology for Java (32-bit and 64-bit)
- Classic VM (64-bit)
- Determining Which JVM to Use
- Capacity Planning
- Tips and Techniques
- Resources

14.1 Introduction

Beginning in V5R4, IBM began a transition to a new VM implementation for i5/OS, IBM Technology for Java, to replace the Classic VM. This transition continues in V6R1 with the introduction of a 64-bit version of IBM Technology for Java, providing a new solution for Java applications which require large amounts of memory. The transition is expected to be completed in the next version of i5/OS, which will no longer support the Classic VM. In the mean time, one of the key performance -related decisions for i5/OS Java users is which JVM to use.

Earlier versions of this document have followed the performance of Java from its infancy to maturity. Early Java applications were often a departure from the traditional OS/400 application architecture, with custom application code responsible for a large portion of the CPU used by the application. Therefore, earlier versions of this document emphasized micro-optimizations – relatively small (though often pervasive) changes to application code to improve performance.

Today's Java applications, however, typically rely on a variety of system services such as JDBC, encryption, and security provided by i5/OS, the Java Virtual Machine (VM), and WebSphere Application Server (WAS), along with other products built on top of WebSphere. As a result, many Java applications now spend far more time in these system services than in custom code. For many applications, this means that performance depends mainly on the performance of IBM code (i5/OS, the Java VM, WebSphere, etc.) and the way that these services are used by the application. Micro-optimizations can still be important in some cases, but are not as critical as they have been in the past.

Tuning is also important for getting good performance in the Java environment. Tuning garbage collection is perhaps the most common example. Thread and connection pool tuning is also frequently important. Proper tuning of i5/OS can also make a big impact on Java application performance.

14.2 What's new in V6R1

In V5R4 IBM introduced IBM Technology for Java, a new VM implementation built on technology used across all of the IBM Systems platforms. In V5R4 only a 32-bit version of IBM Technology for Java was supported; in V6R1, a new 64-bit version of IBM Technology for Java is also available, providing a new

option for Java applications which require large amounts of memory. The Classic VM remains available in V6R1, but future i5/OS releases are expected to support only IBM Technology for Java.

The default VM in V6R1 is IBM Technology for Java 5.0, 32-bit. Other supported versions of IBM Technology for Java include 5.0 64-bit, 6.0 32-bit, and 6.0 64-bit. (6.0 versions will require the latest PTFs to be loaded.) The Classic VM supports Java versions 1.4, 5.0, and 6.0. In V5R4, the default VM is Classic 1.4. Classic 1.3, 5.0, and 6.0 are also supported, as well as IBM Technology for Java 5.0 32-bit and 6.0 32-bit.

Java applications using the Classic VM will generally have equivalent performance between V5R4 and V6R1, although applications which use JDBC to access database may see some improvement. The Classic VM no longer supports Direct Execution (DE) in V6R1; all applications will run with the Just In Time (JIT) compiler. As a result, applications which previously used DE may see some performance difference (usually a significant improvement) when moving to V6R1. Because the same underlying VM is used for all versions of Classic, most applications will see little performance difference between the different JDK levels.

V6R1 offers significant performance improvements over V5R4 when running IBM Technology for Java -- on the order of 10% for many applications, with larger improvements possible when using the -Xlp64k flag to enable 64k pages. In addition, there are substantial performance improvements when moving from IBM Technology for Java 5.0 to 6.0. Performance improvements are frequently introduced in PTFs.

Recent generations of hardware have greatly improved the performance of computationally-intensive applications, which include most Java applications. Since their introduction in V5R3, System i5 servers employing POWER5 processors – models 520, 550, 570, and 595 – have a proven record of providing excellent performance for Java applications. The POWER5+ models introduced with V5R4 build on this success, with performance improvements of up to 30% for the same number of processors in some models. The new POWER6 models introduced in 2007 provide further performance gains, especially for Java applications, which tend to be computationally intensive.

The 515 and 525 models introduced in April, 2007 all include a minimum of 3800 CPW and include L3 cache. These systems deliver solid Java performance at the low-end. Other attractive options at the low-end are the 600 and 1200 CPW models (520-7350 and 520-7352), which have an accelerator feature which allow them to be upgraded to 3100 and 3800 CPW (non-interactive), respectively.

14.3 IBM Technology for Java (32-bit and 64-bit)

IBM's extensive research and development in Java technology has resulted in significant advances in performance and reliability in IBM's Java implementations. Many of these advances have been incorporated into the i5/OS Classic VM, but in order to make the latest developments available to System i customers as quickly as possible, IBM introduced a new 32-bit implementation of Java to i5/OS in V5R4. This VM is built on the same technology as IBM's VMs for other platforms, and provides a modular and flexible base for further improvements in the future. In V6R1, a 64-bit version of the same VM is also available.

IBM Technology for Java currently supports Java 5.0 (JDK version 1.5) and (with the latest PTFs) Java 6 (JDK version 1.6). Older versions of the JDK are only supported with the Classic 64-bit VM.

On i5/OS, IBM Technology for Java runs in i5/OS Portable Application Solutions Environment (i5/OS PASE) with either a 32-bit (for the 32-bit VM) or 64-bit (for the 64-bit VM) environment. Due to sophisticated memory management, both the 32-bit and 64-bit VMs provide a significant reduction in memory requirements over the Classic VM for most applications. Because the 32-bit VM uses only 4 bytes (instead of 8 bytes) for object references, applications will have an even smaller memory footprint with the 32-bit VM; however, the 32-bit address space leads to a maximum heap size of 2.5 - 3.25 GB, which may not be enough memory for some applications.

Because IBM Technology for Java shares a common implementation with IBM's VMs on other platforms, the available tuning parameters are essentially the same on i5/OS as on other platforms. This will require some adjustment for users of the i5/OS Classic VM, but may be a welcome change for those who work with Java on multiple platforms.

Some of the key areas to be aware of when considering use of IBM Technology for Java are described below.

Native Code

Because IBM Technology for Java runs in i5/OS PASE, there is some additional overhead in calls to native ILE code. This may affect performance of certain applications which make calls to native ILE code through the Java Native Interface (JNI). Calls to certain operating system services, such as IFS file access and socket communication, may also have some additional overhead, although the overhead should be minimal for applications with a typical use of these services. Conversely, JNI calls to PASE native methods will have less overhead than they did with the Classic VM, offering a performance improvement for some applications.

The performance impact for JNI method calls to ILE will depend on the frequency of JNI calls and the complexity of the native methods. If the calls are infrequent, or if the native methods are very complex (and therefore take a long time to execute), the increased overhead may not make a big difference in the overall performance of the application. Applications which make frequent calls to simple native methods may see a more significant performance impact compared to the 64-bit Classic VM.

For some applications, it may be possible to port these native methods to run in i5/OS PASE rather than in ILE, greatly reducing the overhead of the native call. In other cases, it may be possible to modify the application to require fewer JNI calls.

Garbage Collection

Recommendations for Garbage Collector (GC) tuning with the i5/OS Classic VM have always been a bit different from tuning recommendations for Java VMs on other platforms. While the main GC tuning parameters (initial and max heap size) have the same names as the key parameters for other VMs, and are set in the same way when running Java from qsh (-Xms and -Xmx), the meaning of these parameters in the Classic 64-bit VM is significantly different. However, with IBM Technology for Java these parameters mean the same thing that they do in IBM VMs on other platforms. Many users will welcome this commonality; however, it does make the transition to the new VM a bit more complicated. The move from a 64-bit VM to a 32-bit VM also complicates matters somewhat, as the ideal heap size will be significantly lower in a 32-bit VM than in a 64-bit VM.

Fortunately, it is not too difficult to come up with parameter values which will provide good performance. If you are moving an application from the Classic VM to IBM Technology for Java, you can use a tool like DMPJVM or verbose GC to determine how large the heap grows when running your application. This value can be used as the maximum heap size for 64-bit IBM Technology for Java; in 32-bit IBM Technology for Java, about 75% of this value is a reasonable starting point. For example, if your application's heap grows to 256 MB when running in the Classic VM, try setting the maximum heap size to 192 MB when running in the 32-bit VM. The initial heap size can be set to about half of this value – 96 MB in our example. These settings are unlikely to provide the best possible performance or the smallest memory footprint, but the application should run reasonably well. Additional performance tests and tuning could result in better settings.

If your application also runs on IBM VMs on other platforms, such as AIX, then you might consider trying the GC parameters from those platforms as a starting point when using IBM Technology for Java on i5/OS.

If you are testing a new application, or aren't certain about the performance characteristics of an existing application running in the Classic 64-bit VM, start by running the application with the default heap size parameters (currently an initial heap size of 4 MB and a maximum of 2 GB). Run the application and see how large the heap grows under a typical peak load. The maximum heap size can be set to this value (or perhaps slightly larger). Then the initial heap size can be increased to improve performance. The optimal value will depend on the application and several other factors, but setting the initial heap size to about 25% of the maximum heap size often provides reasonable performance.

Keep in mind that the maximum heap size for the 32-bit VM is 3328 MB. Attempting to use a larger value for the initial or maximum heap size will result in an error. The maximum heap size is reduced when using IBM Technology for Java's "Shared Classes" feature or when files are mapped into memory (via the java.nio APIs). The maximum heap size can also be impacted when running large numbers of threads, or by the use of native code running in i5/OS PASE, since the memory used by this native code must share the same 32-bit address space as the Java VM. As a result, many applications will have a practical limit of 3 GB (3072 MB) or even less. Applications with larger heap requirements may need to use one of the 64-bit VMs (either IBM Technology for Java or the Classic VM).

When heap requirements are not a factor, the 64-bit version of IBM Technology for Java will tend to be slightly slower (on the order of 10%) than 32-bit with a somewhat larger (on the order of 70%) memory footprint. Thus, the 32-bit VM should be preferred for applications where the maximum heap size limitation is not an issue.

14.4 Classic VM (64-bit)

The 64-bit Classic Java Virtual Machine continues to be supported in V6R1, though most applications should begin migrating to IBM Technology for Java to take advantage of its performance benefits. The integration of the Classic VM into i5/OS provides some unique features and benefits, although this can result in some confusion to users who are familiar with running Java applications on other platforms. Some of the performance-related features you may need to be aware of are described below.

JIT Compiler

Interpreting the platform-neutral bytecodes of a Java class file, bytecode by bytecode, is one valid and robust way to execute Java object code; it is not, however, the fastest way. To approach optimal Java performance, it pays to apply analysis and optimizations to the Java bytecodes, and the resulting machine code.

One approach to optimizing Java bytecode involves analyzing the object code "ahead of time" – before it is actually running. This "ahead-of-time" (AOT) compiler technology was used exclusively by the original AS/400 Java Virtual Machine, whose success proved the power of such an approach.

However, any static AOT analysis suffers one fatal flaw: in a dynamically loading language such as Java, it is impossible for an AOT compiler to know exactly what the environment will look like when the code is actually being executed. Certain valuable optimizations – such as inter-class method inlining or parameter-passing optimizations – cannot be made without adding extra checks to ensure that the optimization is still valid at run-time. While these checks are trimmed down as much as possible, some amount of overhead is unavoidable.

When Java was first introduced to the AS/400 it used an AOT compilation approach, with a combination of bytecode interpretation and Direct Execution (DE) programs to statically optimize Java code for the OS/400 environment, with startup and runtime performance usually significantly faster than what other Java implementations at the time could provide.

Later, "Just-In-Time" (JIT) compiler technology was introduced in many Java VMs. Unlike AOT compilation, JIT compiles Java bytecodes to machine code on-the-fly as the application is running. While this introduces some overhead as the compilation occurs, the compiler can optimize much more aggressively, because it knows the exact state of the system it is compiling for.

Over time, JIT compilation technology improved and was implemented alongside DE in the i5/OS Classic VM. JIT performance overtook DE in the V5R2 time frame for most applications, and has continued to improve at a faster rate. In V6R1, support for DE was eliminated, so the JIT will be used for all Java applications.

Despite the improvements to JIT for both runtime and startup performance, startup time does tend to be slightly longer for JIT than DE. Beginning in V5R2, the Mixed Mode Interpreter (MMI) is used to interpret code until it has been executed a number of times (2000 by default, can be overridden by setting the system property `os400.jit.mmi.threshold`) before JIT compiling it, resulting in improved startup time. V5R3 introduced asynchronous JIT compilation, which further improved startup time, especially on multiprocessor systems. As a result of these and other improvements, many applications will no longer see a significant difference in startup time between DE and JIT. Even if startup time is a bit longer with JIT, the improvement in runtime performance may be worth it, especially for long-running applications which don't start up frequently.

Prior to V6R1, the default execution mode is "`jitc_de`", which uses DE for Java classes which already have DE programs, and JIT for classes which do not. Notably, JDK classes are shipped with DE program objects created, and will therefore use DE by default. Set the system property `java.compiler` to `jitc` to force JIT to be used for all Java code in your application. (See InfoCenter for instructions about setting Java system properties.)

Note that even when running with the JIT, the VM will have to create a Java program object (with optimization level *INTERPRET) the first time a particular Java class is used on the system, if one does not already exist. Creation of this program object is much faster than creating a full DE program, but it

may still make a noticeable difference in startup time the first time your application is used, particularly in applications with a large number of classes. Running CRTJVAPGM with OPTIMIZE(*INTERPRET) will create this program ahead of time, making the first startup faster.

Garbage Collection

Java uses Garbage Collection (GC) to automatically manage memory by cleaning up objects and memory when they are no longer in use. This eliminates certain types of memory leaks which can be caused by application bugs for applications written in other languages. But GC does have some overhead as it determines which objects can be collected. Tuning the garbage collector is often the simplest way to improve performance for Java applications.

The Garbage Collector in the i5/OS Classic VM works differently from collectors in Java VMs on other platforms, and therefore must be tuned differently. There are two parameters that can be used to tune GC: GCHINL (-Xms) and GCHMAX (-Xmx). The JAVA/RUNJVA commands also include GCHPTY and GCHFRQ, but these parameters are ignored and have no effect on performance.

The Garbage Collector runs asynchronously in one or more background threads. When a GC cycle is triggered, the Garbage Collector will scan the entire Java heap, and mark each of the objects which can still be accessed by the application. At the end of this "mark" phase, any objects which have not been marked are no longer accessible by the application, and can be deleted. The Garbage Collector then "sweeps" the heap, freeing the memory used by all of these inaccessible objects.

A GC cycle can be triggered in a few different ways. The three most common are:
1. An amount of memory exceeding the collection threshold value (GCHINL) has been allocated since the previous GC cycle began.
2. The heap size has reached the maximum heap value (GCHMAX).
3. The application called *java.lang.System.gc( )* [not recommended for most applications]

The collection threshold value (GCHINL or -Xms, often referred to as the "initial heap size") is the most important value to tune. The default size for V5R3 and later is 16 MB. Using larger values for this parameter will allow the heap to grow larger, which means that GC will run less frequently, but each cycle will take longer. Using smaller values will keep the heap smaller, but GC will run more often. The best value depends on the number, size, and lifetime of objects in your application as well as the amount of memory available to the application. Most applications will benefit from using a larger collection threshold value – 96 MB is reasonable for many applications. For WebSphere applications on larger systems, heap threshold values of 512 MB or more are not uncommon.

The maximum heap size (GCHMAX, or -Xmx) specifies the largest that the heap is allowed to grow. If the heap reaches this size, a synchronous garbage collection will be performed. All other application threads will have to wait while this GC cycle occurs, resulting in longer response times. If this synchronous GC cycle is not able to free up enough memory for the application to continue, an OutOfMemoryError will be thrown. The default value for this parameter is *NOMAX, meaning that there is no limit to the heap size. In practice, a well behaved application will settle out to some steady state heap size, so *NOMAX does not mean that the heap will grow infinitely large. Most applications can leave this parameter at its default value.

One important consideration is to not allow the Java heap to grow beyond the amount of physical memory available to the application. For example, if the application is running in the *BASE memory pool with a size of 1 GB, and the heap grows to 1.5 GB, the paging rate will tend to get quite high, especially when a

GC cycle is running. This will show up as non-database page faults on the WRKSYSSTS command display; rates of 20 to 30 faults per second are usually acceptable, but larger values may indicate a performance problem. In this case, the size of the memory pool should be increased, or the collection threshold value (GCHINL or -Xms) should be decreased so the heap isn't allowed to grow as large. In many cases the scenario may be complicated by the fact that multiple applications may be running in the same memory pool. Therefore, the total memory requirements of all of these applications must be considered when setting the pool size. In some environments it may be useful to run key Java applications in a private pool in order to have more control over the memory available to these applications.

In some cases it may also be helpful to set the maximum heap size to be slightly larger than the memory pool size. This will act as a safety net so that if the heap does grow beyond the memory pool size, it will not cause high paging rates. In this case, the application will probably not be usable (due to the synchronous garbage collection cycles and OutOfMemoryErrors that may occur), but it will have less impact on any other applications running on the system.

A final consideration is the application's use of objects. While the garbage collector will prevent certain types of memory leaks, it is still possible for an application to have an "object leak". One common example is when the application adds new objects to a List or Map, but never removes the objects. Therefore the List or Map continues to grow, and the heap size grows along with it. As this growth continues, the garbage collector will begin taking longer to run each cycle, and eventually you may exhaust the physical memory available to the application. In this case, the application should be modified to remove the objects from the List or Map when they are no longer needed so the heap can remain at a reasonable size. A similar example involves the use of caches inside the application. If these caches are allowed to grow too large, they may consume more memory than is physically available on the system. Using smaller cache sizes may improve the performance of your application.

Bytecode Verification

In order to maintain system stability and security, it is important that Java bytecodes are verified before they are executed, to ensure that the bytecodes don't try to do anything not allowed by the Java VM specification. This verification is important for any Java implementation, but especially critical for server environments, and perhaps even more so on i5/OS where the JVM is integrated into the operating system. Therefore, in i5/OS, bytecode verification is not only turned on by default, but it is impossible to turn it off. While the bytecode verification step isn't especially slow, it can impact startup time in certain cases – especially when compared to VMs on other platforms which may not do bytecode verification by default. In most cases, full bytecode verification can be done just once for a class, and the resulting JVAPGM objects saved with its corresponding class or jar file as long as the class doesn't change.

However, when user classloaders are used to load classes, the VM may not be able to locate the file from which the class was loaded (in particular, if the standard URLClassLoader mechanism is not being used by the user classloader). In this case, the bytecode verification cache is used to minimize the cost of bytecode verification.

The verification cache operates by caching JVAPGMs that have been dynamically created for dynamically loaded classes. When the verification cache is not operating, these JVAPGMs are created as temporary objects, and are deleted as the JVM shuts down. When the verification cache is enabled, however, these JVAPGMs are created as persistent objects, and are cached in the (user specified) machine-wide cache file. If the same (byte-for-byte identical) class is dynamically loaded a second time (even after the machine is re-IPLed), the cached JVAPGM for that class is located in the cache and

reused, eliminating the need to verify the class and create a new JVAPGM (and eliminating the time and performance impact that would be required for these actions). Older JVAPGMs are "aged out" of the cache if they are not used within a given period of time (default is one week).

In general, the only cost of enabling the verification cache is a modest amount of disk space. If it turns out that your application is not using one of the problem user class loaders, the cache will have no impact, positive or negative, while if your application is using such a class loader then the time taken to create and cache the persistent JVAPGM is only slightly more than the time required to create a temporary JVAPGM. With next to zero downside risk, and a decent potential to improve performance, the verification cache is well worth a try.

Maintenance is not a problem either: if the source for a cached JVAPGM is changed, the currently-cached version will simply "age out" (since its class will no longer be a byte-for-byte match), and a new JVAPGM will be silently created and cached. Likewise, the cache doesn't care about JDK versions, PTFs installed, application upgrades, etc.

14.5 Determining Which JVM to Use

Beginning in V5R4, applications can run in either the Classic 64-bit VM or with IBM Technology for Java (32-bit only in V5R4, 32-bit or 64-bit in V6R1). Both VM implementations provide a fully compliant implementation of the Java specifications, and pure Java applications should be able to run without changes in either VM by setting the JAVA_HOME environment variable appropriately. (See InfoCenter for details on specifying which VM will be used to execute a Java program.) However, some applications may have dependencies which will prevent them from working on one of the VM implementations.

In general, applications should use 32-bit IBM Technology for Java when possible. Applications which require larger heaps than can be managed with a 32-bit VM should use 64-bit IBM Technology for Java (on V6R1). The Classic VM also remains available for cases where IBM Technology for Java is not appropriate and to ease migration from older releases.

Some factors to consider include:

Functional Considerations

1. The Classic VM is not supported in IBM i 7.1. Only IBM Technology for Java 32-bit and 64-bit JVMs are supported.
2. IBM Technology for Java was introduced in i5/OS V5R4M0. Older versions of OS/400 and i5/OS support only the Classic VM.
3. IBM Technology for Java only supports Java 5.0 (JDK 1.5) and higher. Older versions of Java (1.4, 1.3, etc.) are not supported. While the Java versions are generally backward compatible, some libraries and environments may require a particular version. The Classic VM continues to support JDK 1.3, 1.4, 1.5 (5.0), and 1.6 (6.0) in V5R4, and JDK 1.4, 1.5 (5.0), and 1.6 (6.0) in V6R1.
4. The Classic VM supported an i5/OS-specific feature called Adopted Authority. IBM Technology for Java does not support this feature, so applications which require Adopted Authority must run in the Classic VM. This will not affect most applications. Applications which do use Adopted Authority should consider migrating to APIs in IBM Toolbox for Java which can serve a similar purpose.
5. Java applications can call native methods through the Java Native Interface (JNI) with either VM. When using IBM Technology for Java, these native programs must be compiled with teraspace

storage enabled. In addition, whenever a buffer is passed to JNI functions such as *GetxxxArrayRegion*, the pointer must point to teraspace storage.

6. When using 32-bit IBM Technology for Java runs in a 32-bit PASE environment, any PASE native methods must also be 32-bit. With 64-bit IBM Technology for Java, PASE native methods must be 64-bit. The Classic VM can call both 32-bit and 64-bit PASE native methods. All of the VMs can call ILE native methods as well.

Performance Considerations

1. When properly tuned, applications will tend to use significantly less memory when running in IBM Technology for Java than in the Classic VM. Performance tests have shown a reduction of 40% or more in the Java heap for most applications when using the 32-bit IBM Technology for Java VM, primarily because object references are stored with only 4 bytes (32 bits) rather than 8 bytes (64 bits). Therefore, an application using 512 MB of heap space in the 64-bit Classic VM might require 300 MB or even less when running in 32-bit IBM Technology for Java. The difference between the Classic VM and 64-bit IBM Technology for Java is somewhat less noticeable, but 64-bit IBM Technology for Java will still tend to have a smaller footprint than Classic for most applications.
2. The downside to using a 32-bit address space is that it limits the amount of memory available to the application. As discussed above, the 32-bit VM has a maximum heap size of 3328 MB, although most applications will have a practical limit of 3 GB or less. Applications which require a larger heap should use 64-bit IBM Technology for Java or the Classic VM. Since applications will use less memory when running in the 32-bit VM, this means that applications which require up to about 5 GB in the Classic VM will probably be able to run in the 32-bit VM. Of course, applications with heap requirements near the 3 GB limit will require extra testing to ensure that they are able to run properly under full load over an extended period of time.
3. Applications which use a single VM to fully utilize large systems (especially 8-way and above) will tend to require larger heap sizes, and therefore may not be able to use the 32-bit VM. In some cases it may be possible to divide the work across two or more VMs. Otherwise, it may be necessary to use one of the 64-bit VMs on large systems to allow larger heap sizes.
4. Because calls to native ILE code are more expensive in IBM Technology for Java, extra care should be taken when moving Java applications which make heavy use of native ILE code to the new VM. Performance testing should be performed to determine whether or not the overhead of the native ILE calls are hurting performance in your application. If this is an issue, the techniques discussed above should be used to attempt to improve the performance. If the performance is still unacceptable, it may be best to continue using the Classic VM at this time. Conversely, applications which make use of i5/OS PASE native methods may see a performance improvement when running in IBM Technology for Java due to the reduced overhead of calling i5/OS PASE methods.
5. Remember that microbenchmarks (small tests to exercise a specific function) do not provide a good measure of performance. Comparisons between the IBM Technology for Java and Classic based on microbenchmarks will not give an accurate picture of how your application will perform in the two VMs, because your application will have different characteristics than the microbenchmark. The best way to determine which VM provides the best performance for your application is to test with the application itself or a reasonably complete subset of the application, using a load generating tool to simulate a load representative of your planned deployment environment.

WebSphere applications running with IBM Technology for Java will be subject to the same constraints as plain Java applications; however, there are some considerations which are specific to WebSphere, as described in Chapter 15 (Web Server and WebSphere Performance).

14.6 Capacity Planning

Due to the wide variety of Java applications which can be developed, it is impossible to make precise capacity planning recommendations which would apply to all applications. It is possible, however, to make some general statements which will apply to most applications. Determining specific system requirements for a particular application requires performance testing with that application. The Workload Estimator can also be used to assist with capacity planning for specific environments, such as WebSphere Application Server or WebSphere Commerce applications.

Despite substantial progress at the language execution level, Java continues to require, on average, processors with substantially higher capabilities than the same machine primarily running RPG and COBOL. This is partially due to the overhead of using an object oriented, garbage collected language. But perhaps more important is that Java applications tend to do more than their counterparts written in more traditional languages. For example, Java applications frequently include more network access and data transformation (like XML) than the RPG and COBOL applications they replace. Java applications also typically use JDBC with SQL to access the database, while traditional iSeries applications tend to use less expensive data access methods like Record Level Access. Therefore, Java applications will continue to require more processor cycles than applications with "similar" functionality written in RPG or COBOL.

As a result, some models at the low end may be suitable for traditional applications, but will not provide acceptable performance for applications written in Java.

General Guidelines

- Remember to account for non-Java work on the system. Few System i servers are used for a single application; most will have a combination of Java and non-Java applications running on the system. Be sure to factor in capacity requirements for both the Java and the non-Java applications which will run on the system. The eServer Workload Estimator can be used to estimate system requirements for a variety of application types.

- Similarly, be sure to consider additional system services which will be used when adding a new Java application to the system. Most Java applications will make use of system services like network communications and database, which may require additional system resources. In particular, the use of JDBC and dynamic SQL can increase the cost of database access from Java compared to traditional applications with similar function.

- Also consider which applications on the system are likely to experience future growth, and adjust the system requirements accordingly. For example, if a Java/WebSphere application is used as the core of an e-business application, then it may see significantly more growth (requiring additional system resources) over time or during particular times of the year than other applications on the system.

- Beware of misleading benchmarks. Many benchmarks are available to test Java performance, but most of these are not good predictors of server-side Java performance. Some of these benchmarks are single-threaded, or run for a very short period of time. Others will stress certain components of the JVM heavily, while avoiding other functionality that is more typical of real applications. Even the best benchmarks will exercise the JVM differently than real applications with real data. This doesn't mean that benchmarks aren't useful; however, results from these benchmarks must be interpreted carefully.

- 5250 OLTP isn't needed for Java applications, although some Java applications will execute 5250 operations that do require 5250 OLTP. Again, be sure to account for non-Java workloads on the system that do require 5250 OLTP.

- Java applications are inherently multi-threaded. Even if the application itself runs in a single thread, VM functionality like Garbage Collection and asynchronous JIT compilation will run in separate threads. As a result, Java will tend to benefit from processors which support Simultaneous Multi-threading (SMT). See Chapter 19 for additional information on SMT. Java applications may also benefit more from systems with multiple processors than single-threaded traditional applications, as multiple application threads can be running in parallel.

- Java tends to require more main storage (memory) than other languages, especially when using the Classic VM. The 64-bit VMs (both Classic and IBM Technology for Java) will also tend to require more memory than is needed by 32-bit VMs on other platforms.

- Along the same lines, Java applications generally benefit more from L3 cache than applications in other languages. Therefore, Java performance may scale better than CPW ratings would indicate when moving from a system with no L3 cache to a system that does have L3 cache. Conversely, Java performance on a system without L3 cache may be worse than the CPW rating suggests. See Appendix C of this document for information on which systems include L3 cache.

- DASD (hard disk) requirements typically don't change much for Java applications compared to applications written in languages like RPG. The biggest use of DASD is usually database, and database sizes do not inherently change when running Java.

14.7 Java Performance – Tips and Techniques

Introduction

Tips and techniques for Java fall into several basic categories:

1. i5/OS Specific. These should be checked out first to ensure you are getting all you should be from your i5/OS Java application.

2. Classic VM Specific. Many i5/OS-specific tips apply only when using the Classic VM and not for IBM Technology for Java.

3. Java Language Specific. Coding tips that will ordinarily improve any Java application, or especially improve it on i5/OS.

4. Database Specific. Use of database can invoke significant path length in i5/OS. Invoking it efficiently can maximize the performance and value of a Java application.

i5/OS Specific Java Tips and Techniques

- *Load the latest CUM package and PTFs*
  To be sure that you have the best performing code, be sure to load the latest CUM packages and PTFs

for all products that you are using. In particular, performance improvements are often introduced in new Java Group PTFs (SF99269 for V5R3, SF99291 for V5R4, and SF99562 for V6R1).

- *Explore the General Performance Tips and Techniques in Chapter 19*
  Some of the discussion in that chapter will apply to Java. Pay particular attention to the discussion "Adjusting Your Performance Tuning for Threads." Specifically, ensure that MAXACT is set high enough to allow all Java threads to run.

- *Consider running Java applications in a separate memory pool*
  On systems running multiple workloads simultaneously, putting Java applications in their own pool will ensure that the Java applications have enough memory allocated to them.

- *Make sure SMT is enabled on systems that support it*
  Java applications are always multi-threaded, and should benefit from Simultaneous Multi-threading (SMT). Ensure that it is turned on by setting the system value QPRCMLTTSK to 1 (On). See chapter 19 for additional details on SMT.

- *Avoid starting new Java VMs frequently*
  Starting a new VM (e.g. through the JAVA/RUNJVA commands) is expensive on any platform, but perhaps a bit more so on i5/OS, due to the relatively high cost of starting a new job. Other factors which make Java startup slow include class loading, bytecode verification, and JIT compilation. As a result, it is far better to use long-running Java programs rather than frequently starting new VMs. If you need to invoke Java frequently from non-Java programs, consider passing messages through an i5/OS Data Queue. The ToolBox Data Queue classes may be used to implement "hot" VM's.

Classic VM-specific Tips

- *Use java.compiler=jitc*
  The JIT compiler now outperforms Direct Execution for nearly all applications. Therefore, java.compiler=jitc should be used for most Java applications. One possible exception is when startup time is a critical issue, but JIT may be appropriate even in these cases. Setting `java.compiler` is not necessary for Classic on V6R1, or for IBM Technology for Java on either V5R4 or V6R1 -- the JIT compiler is always used in these cases.

- *Delete existing DE program objects*
  When using the JIT, JVAPGM objects containing Direct Execution machine code are not used. These program objects can be large, so removing the unused JVAPGM objects can free up disk space. This is not needed on V6R1. To determine if your class/zip/jar file has a permanent, hidden program object on previous releases, use the DSPJVAPGM command. If a Java program is associated with the file, and the “Optimization” level is something other than *INTERPRET, use DLTJVAPGM to delete the hidden program. DLTJVAPGM does not affect the jar or zip file itself; only the hidden program. Do not use DLTJVAPGM on IBM-shipped JDK jar files (such as rt.jar). As explained earlier, the JIT does take advantage of programs created at optimization *INTERPRET. These programs require significantly less space and do not need to be deleted. Program objects (even at *INTERPRET) are not used by IBM Technology for Java.

- *Consider the special property os400.jit.mmi.threshold.*
  This property sets the threshold for the MMI of the JIT. Setting this to a small value will result is compilation of the classes at startup time and will increase the start up time. In addition, using a very small value (less than 50) may result in a slower compiled version, since profiling data gathered

during the interpreted phase may not be representative of the actual application characteristics. Setting this to a high value may result in a somewhat faster startup time and compilation of the classes will occur once the threshold is reached. However, if the value is set too high then an increased warm-up time may occur since it will take additional time for the classes to be optimized by the JIT compiler.

The default value of 2000 is usually OK for most scenarios. This property has no effect when using IBM Technology for Java.

- *Package your Java application as a .jar or .zip file.*
  Packaging multiple classes in one .zip or .jar file should improve class loading time and also code optimization when using Direct Execution (DE). Within a .zip or .class file, i5/OS Java will attempt to in-line code from other members of the .zip or .jar file.

Java Language Performance Tips

Due to advances in JIT technology, many common code optimizations which were critical for performance a few years ago are no longer as necessary in modern JVMs. Even today, these techniques will not hurt performance. But they may not make a big positive difference either. When making these types of optimizations, care should be taken to balance the need for performance with other factors such as code readability and the ease of future maintenance. It is also important to remember that the majority of the application's CPU time will be spent in a small amount of code. CPU profiling should be used to identify these "hot spots", and optimizations should be focused on these sections of code.

Various Java code optimizations are well documented. Some of the more common optimizations are described below:

- *Minimize object creation*
  Excessive object creation is a common cause of poor application performance. In addition to the cost of allocating memory for the new object and invoking its constructor, the new object will use space in the Java heap, which will result in longer garbage collection cycles. Of course, object creation cannot be avoided, but it can be minimized in key areas.

  The most important areas to look at for reducing object creation is inside loops and other commonly-executed code paths. Some common causes of object creation include:

  - *String.substring*( ) creates a new String object.
  - The arithmetic methods in java.math.BigDecimal (*add*, *divide*, etc) create a new BigDecimal object.
  - The I/O method *readLine*( ) (e.g. in java.io.BufferedReader) will create a new String.
  - String concatenation (e.g.: "The value is: " + value) will generally result in creation of a StringBuffer, a String, and a character array.
  - Putting primitive values (like *int* or *long*) into a collection (like *List* or *Map*) requires wrapping it in a new object (e.g. Java.lang.Integer). This is usually obvious in the code, but Java 5.0 introduced the concept of *autoboxing* which will perform this wrapping automatically, hiding the object creation from the programmer.

Some objects, like StringBuffer, provide a way to reset the object to its initial state, which can be useful for avoiding object creation, especially inside loops. For StringBuffer, this can be done by calling *setLength(0)*.

- *Minimize synchronized methods*
  Synchronized methods/blocks can have significantly more overhead than non-synchronized code. This includes some overhead in acquiring locks, flushing caches to correctly implement the Java memory model, and contention on locks when multiple threads are trying to hold the same lock at the same time. From a performance standpoint, it is best if synchronized code can be avoided. However, it is important to remember that improperly synchronized code can cause functional or data-integrity issues; some of these issues may be difficult to debug since they may only occur under heavy load. As a result, it is important to ensure that changes to synchronization are "safe". In many cases, removing synchronization from code may require design changes in the application.

  Some common synchronization patterns are easily illustrated with Java's built-in String classes. Most other Java classes (including user-written classes) will follow one of these patterns. Each has different performance characteristics.

  - java.lang.String is an *immutable* object – once constructed, it cannot be changed. As a result, it is inherently thread-safe and does not require synchronization. However, since Strings cannot be modified, operations which require a modified String (like *String.substring()*) will have to create a new String, resulting in more object creation.

  - java.lang.StringBuffer is a mutable object which can change after it is constructed. In order to make it thread-safe, nearly all methods in the class (including some which do not modify the StringBuffer) are synchronized.

  - java.lang.StringBuilder (introduced in Java 5.0) is an unsynchronized version of StringBuffer. Because its methods are not synchronized, this class is not thread-safe, so StringBuilder instances can not be shared between threads without external synchronization.

  Dealing with synchronization correctly requires a good understanding of Java and your application, so be careful about applying this tip.

- *Use exceptions only for "exceptional" conditions*
  The "try" block of an exception handler carries little overhead. However, there is significant overhead when an exception is actually thrown and caught. Therefore, you should use exceptions only for "exceptional" conditions; that is, for conditions that are not likely to happen during normal execution. For example, consider the following procedure:

```
public void badPrintArray (int arr[]) {
    int i = 0;
    try {
        while (true) {
            System.out.println (arr[i++]);
        }
    } catch (ArrayOutOfBoundsException e) {
        // Reached the end of the array....exit
    }
}
```

  Instead, the above procedure should be written as:

```
public void goodPrintArray (int arr[]) {
    int len = arr.length;
    for (int i = 0; i < len; i++) {
        System.out.println (arr[i]);
    }
}
```

In the “bad” version of this code, an exception will always be thrown (and caught) in every execution of the method. In the “good” version, most calls to the method will not result in an exception. However, if you passed “null” to the method, it would throw a NullPointerException. Since this is probably not something that would normally happen, an exception may be appropriate in this case. (On the other hand, if you expect that null will be passed to this method frequently, it may be a good idea to handle it specifically rather than throwing an exception.)

- *Use static final when creating constants*
  When data is invariant, declare it as static final. For example here are two array initializations:

```
class test1 {
      int myarray[] =
                { 1,2,3,4,5,6,7,8,9,10,
                  2,3,4,5,6,7,8,9,10,11,
                  3,4,5,6,7,8,9,10,11,12,
                  4,5,6,7,8,9,10,11,12,13,
                  5,6,7,8,9,10,11,12,13,14 };
}

class test2 {
      static final int myarray2[] =
                  { 1,2,3,4,5,6,7,8,9,10,
                    2,3,4,5,6,7,8,9,10,11,
                    3,4,5,6,7,8,9,10,11,12,
                    4,5,6,7,8,9,10,11,12,13,
                    5,6,7,8,9,10,11,12,13,14 };
}
```

  Since the array myarray2 in class test2 is defined as *static*, there is only one myarray2 array for all the many creations of the test2 object. In the case of the test1 class, there is an array myarray for *each* test1 instance. The use of *final* ensures that the array cannot be changed, making it safe to use from multiple threads.

Java i5/OS Database Access Tips

- *Use the native JDBC driver*
  There are two i5/OS JDBC drivers that may be used to access local data: the Native driver (using a JDBC URL *"jdbc:db2:system-name"*) and the Toolbox driver (with a JDBC URL *"jdbc:as400:system-name"*). The native JDBC driver is optimized for local database access, and gives the best performance when accessing the database on the same system as your Java applications. The Toolbox driver supports remote access, and should be used when accessing the database on a separate system. This recommendation is true for both the 64-bit Classic VM and the new 32-bit VM.

- *Pool Database Connections*
  Connection pooling is a technique for sharing a small number of database connections among a number of threads. Rather than each thread opening a connection to the database, executing some requests, and then closing the connection, a connection can be obtained from the connection pool,

used, and then returned to the pool. This eliminates much of the overhead in establishing a new JDBC connection. WebSphere Application Server uses built-in connection pooling when getting a JDBC connection from a DataSource.

- *Use Prepared Statements*
  The JDBC *prepareStatement* method should be used for repeatable *executeQuery* or *executeUpdate* methods. If *prepareStatement*, which generates a reusable PreparedStatement object, is not used, the *execute* statement will implicitly re-do this work on every *execute* or *executeQuery*, even if the query is identical. WebSphere's DataSource will automatically cache your PreparedStatements, so you don't have to keep a reference to them – when WebSphere sees that you are attempting to prepare a statement that it has already prepared, it will give you a reference to the already prepared statement, rather than creating a new one. In non-WebSphere applications, it may be necessary to explicitly cache PreparedStatement objects.

  When using PreparedStatements, be sure to use parameter markers for variable data, rather than dynamically building query strings with literal data. This will enable reuse of the PreparedStatement with new parameter values.

  Avoid placing the prepareStatement inside of loops (e.g. just before the execute). In some non-i5/OS environments, this just-before-the-query coding practice is common for non-Java languages, which required a "prepare" function for any SQL statement. Programmers may carry this practice over to Java. However, in many cases, the prepareStatement contents don't change (this includes parameter markers) and the Java code will run faster on all platforms if it is executed only one time, instead of once per loop. This technique may show a greater improvement on i5/OS.

- *Store or at least fetch numeric data in DB2 as double*
  Fixed-precision decimal data cannot be represented in Java as a primitive type. When accessing numeric and decimal fields from the database through JDBC, values can be retrieved using getDouble( ) or getBigDecimal( ). The latter method will create a new java.math.BigDecimal object each time it is called. Using getDouble (which returns a primitive double) will give better performance, and should be preferred when floating-point values are appropriate for your application (i.e. for most applications outside the financial industry).

- *Consider using ToolBox record I/O*
  The IBM Toolbox for Java provides native record level access classes. These classes are specific to the i5/OS platform. They may provide a significant performance gain over the use of JDBC access for applications where portability to other databases is not required. See the AS400File object under Record Level access in the InfoCenter.

Resources

The i5/OS Java and WebSphere performance team maintains a list of performance-related documents at http://www.ibm.com/systems/i/solutions/perfmgmt/webjtune.html.

The Java Diagnostics Guide provides detailed information on performance tuning and analysis when using IBM Technology for Java. Most of the document applies to all platforms using IBM's Java VM; in addition, one chapter is written specifically for i5/OS information. The Diagnostics Guide is available at http://www.ibm.com/developerworks/java/jdk/diagnosis/.

Chapter 15. Web Server and WebSphere Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

This section discusses IBM i performance information in Web serving and WebSphere environments. Specific products that are discussed include: HTTP Server (powered by Apache) (in section 15.1), PHP - Zend Core for i (15.2), WebSphere Application Server (15.3), Web Facing (15.4), Host Access Transformation Services (15.5), WebSphere Portal Server (15.6), WebSphere Commerce (15.7), WebSphere Commerce Payments (15.8), and WebSphere MQ (15.9).

The primary focus of this section will be to discuss the performance characteristics of the System i platform as a server in a Web environment, provide capacity planning information, and recommend actions to help achieve high performance. Having a high-performance network infrastructure is very important for Web environments; please refer to Chapter 2, "Communications Performance" for related information and tuning tips.

Web Overview: There are many factors that can impact overall performance (e.g., end-user response time, throughput) in the complex Web environment, some of which are listed below:

1) Web Browser or client
- processing speed of the client system
- performance characteristics and configuration of the Web browser
- client application performance characteristics

2) Network
- speed of the communications links
- capacity and caching characteristics of any proxy servers
- the responsiveness of any other related remote servers (e.g., payment gateways)
- congestion of network resources

3) IBM i Web Server and Applications
- IBM i processor capacity (indicated by the CPW value)
- utilization of key System i server resources (CPU, IOP, memory, disk)
- Web server performance characteristics
- application (e.g., CGI, servlet) performance characteristics

Comparing traditional communications to Web-based transactions: For commercial applications, data accesses across the Internet differs distinctly from accesses across 'traditional' communications networks. The additional resources to support Internet transactions by the CPU, IOP, and line are significant and must be considered in capacity planning. Typically, in a traditional network:
- there is a request and response (between client and server)
- connections/sessions are maintained between transactions
- networks are well-understood and tuned

Typically for Web transactions, there may be a dozen or more line transmissions per transaction:
- a connection is established/closed for each transaction
- there is a request and response (between client and server)
- one user transaction may contain many separate Internet transactions
- secure transactions are more frequent and consume more resource

- with the Internet, the network may not be well-understood (route, components, performance)

Information source and disclaimer: The information in the sections that follow is based on performance measurements and analysis done in the internal IBM performance lab. The raw data is not provided here, but the highlights, general conclusions, and recommendations are included. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here. Note that these workloads are measured in best-case environments (e.g., local LAN, large MTU sizes, no errors). Real Internet networks typically have higher contention, higher levels of logging and security, MTU size limitations, and intermediate network servers (e.g., proxy, SOCKS); and therefore, it would likely consume more resources.

15.1 HTTP Server (powered by Apache)

The HTTP Server (powered by Apache) for i5/OS has some exciting new features for V5R4. The level of the HTTP Server has been increased to support Apache 2.0.52 and is now a UTF-8 server. This means that requests are being received and then processed as UTF-8 rather than first being converted to EBCDIC and then processed. This will make porting open source modules for the HTTP Server on your IBM System i easier than before. For more information on what's new for HTTP Server for i5/OS, visit http://www.ibm.com/servers/eserver/iseries/software/http/news/sitenews.html

This section discusses some basic information about HTTP Server (powered by Apache) and gives you some insight about the relative performance between primitive HTTP Server tests.

The typical high-level flow for Web transactions: the connection is made, the request is received and processed by the HTTP server, the response is sent to the browser, and the connection is ended. If the browser has multiple file requests for the same HTTP server, it is possible to get the multiple requests with one connection. This feature is known as *persistent connection* and can be set using the KeepAlive directive in the HTTP server configuration.

To understand the test environment and to better interpret performance tools reports or screens it is helpful to know that the following jobs and tasks are involved: communications router tasks (IPRTRnnn), several HTTP jobs with at least one with many threads, and perhaps an additional set of application jobs/threads.

"Web Server Primitives" Workload Description: The "Web Server Primitives" workload is driven by the program ApacheBench 2.0.40-dev that runs on a client system and simulates multiple Web browser clients by issuing URL requests to the Web Server. The number of simulated clients can be adjusted to vary the offered load, which was kept at a moderate level. Files and programs exist on the IBM System i platform to support the various transaction types. Each of the transaction types used are quite simple, and will serve a static response page of specified data length back to the client. Each of the transactions can be served in a secure (HTTPS:) or a non-secure (HTTP:) fashion. The HTTP server environment is a partition of an IBM System i 570+ 8-Way (2.2Ghz), configured with one dedicated CPU and a 1 Gbps communication adapter.

- **Static Page:** HTTP retrieves a file from IFS and serves the static page. The HTTP server can be configured to cache the file in its local cache to reduce server resource consumption. FRCA (Fast Response Caching Accelerator) can also be configured to cache the file deeper in the operating system and further reduce resource consumption.

- **CGI**: HTTP invokes a CGI program which builds a simple HTML page and serves it via the HTTP server. This CGI program can run in either a new or a named activation group. The CGI programs were compiled using a "named" activation group unless specified otherwise.

Web Server Capacity Planning: Please use the IBM Systems Workload Estimator to do capacity planning for Web environments using the following workloads: Web Serving, WebSphere, WebFacing, WebSphere Portal Server, WebSphere Commerce. This tool allows you to suggest a transaction rate and to further characterize your workload. You'll find the tool along with good help text at: http://www.ibm.com/systems/support/tools/estimator . Work with your marketing representative to utilize this tool (also chapter 20).

The following tables provide a summary of the measured performance data for both static and dynamic Web server transactions. These charts should be used in conjunction with the rest of the information in this section for correct interpretation. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here.

Relative Performance Metrics:

- *"Relative Capacity Metric:* This metric is used throughout this section to demonstrate the relative capacity performance between primitive tests. Because of the diversity of each environment the ability to scale these results could be challenging, but they are provided to give you an insight into the relation between the performance of each primitive HTTP Server test.

Table 15.1 i5/OS V5R4 Web Serving Relative Capacity - Static Page

Transaction Type:	Relative Capacity Metrics	
	Non-secure	**Secure**
Static Page - IFS	2.016	1.481
Static Page - Local Cache	3.538	2.235
Static Page - FRCA	34.730	n/a

Notes/Disclaimers:
- Data assumes no access logging, no name server interactions, KeepAlive on, LiveLocalCache off
- Secure: 128-bit RC4 symmetric cipher and MD5 message digest with 1024-bit RSA public/private keys
- These results are relative to each other and do not scale with other environments
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.

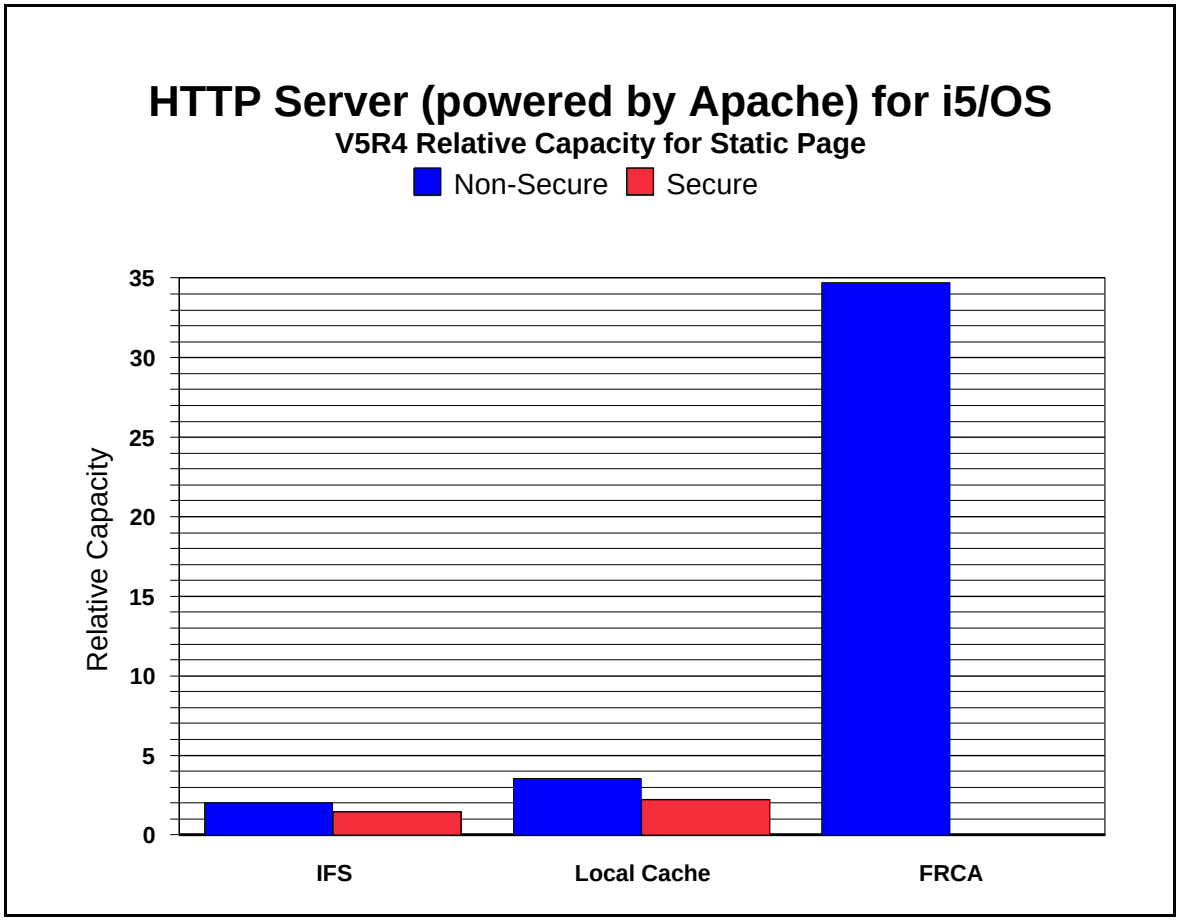


Figure 15.1 i5/OS V5R4 Web Serving Relative Capacities - Various Transactions

Table 15.2 i5/OS V5R4 Web Serving Relative Capacity - CGI

Transaction Type:	Relative Capacity Metrics	
	Non-secure	**Secure**
CGI - New Activation	0.092	0.090
CGI - Named Activation	0.475	0.436

Notes/Disclaimers:
- Data assumes no access logging, no name server interactions, KeepAlive on, LiveLocalCache off
- Secure: 128-bit RC4 symmetric cipher and MD5 message digest with 1024-bit RSA public/private keys
- These results are relative to each other and do not scale with other environments
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.

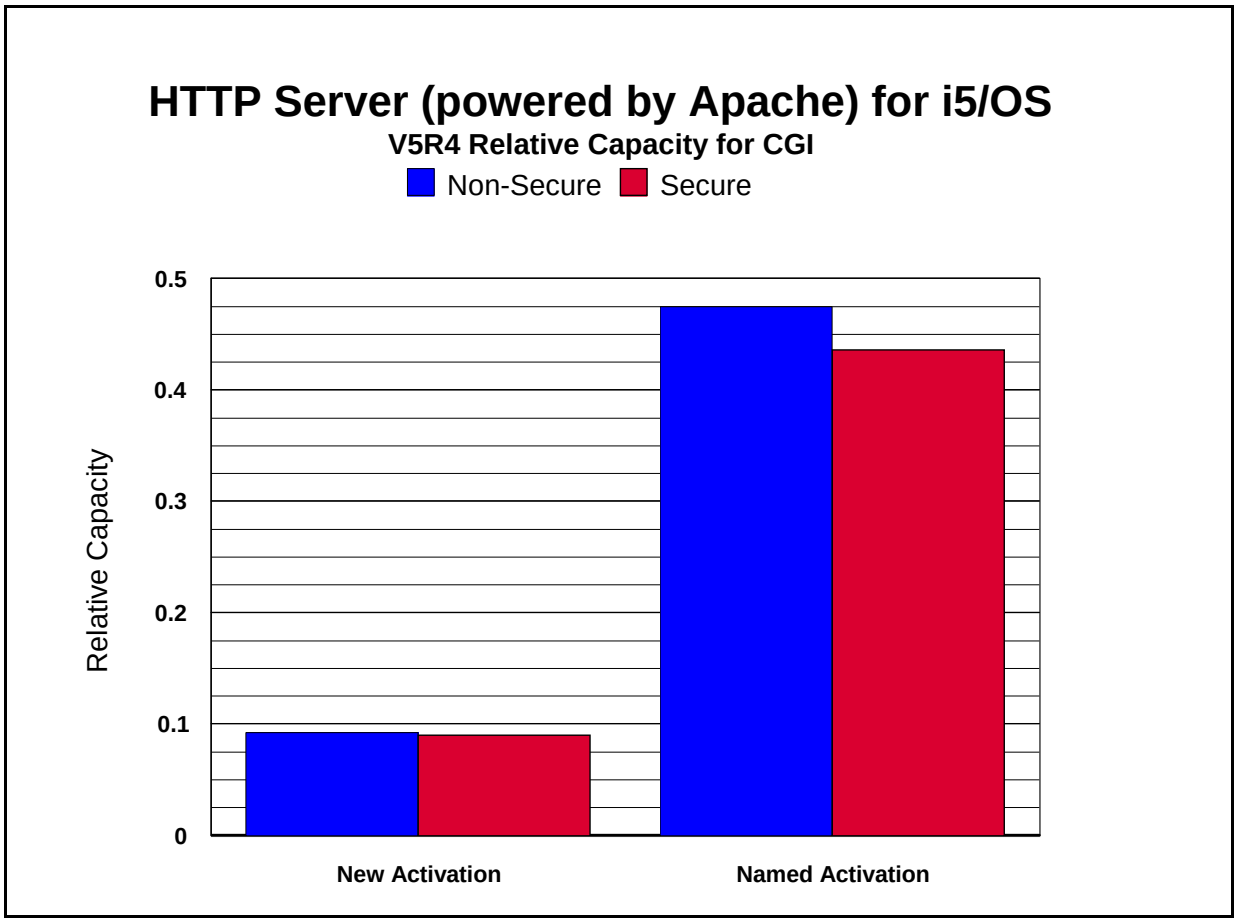


Figure 15.2 i5/OS V5R4 Web Serving Relative Capacities - Various Transactions

Table 15.3 i5/OS V5R4 Web Serving Relative Capacity for Static (varied sizes)

	Relative Capacity Metrics					
Transaction Type:	**1K Bytes**		**`10K Bytes**		**100K Bytes**	
KeepAlive	**Off**	**On**	**Off**	**On**	**Off**	**On**
Static Page - IFS	1.558	2.016	1.347	1.793	0.830	1.068
Static Page - Local Cache	2.407	3.538	2.095	3.044	0.958	1.243
Static Page - FRCA	11.564	34.730	7.691	13.539	1.873	2.622

Notes/Disclaimers:
- These results are relative to each other and do not scale with other environments.
- IBM System i CPU features without an L2 cache will have lower web server capacities than the CPW value would indicate

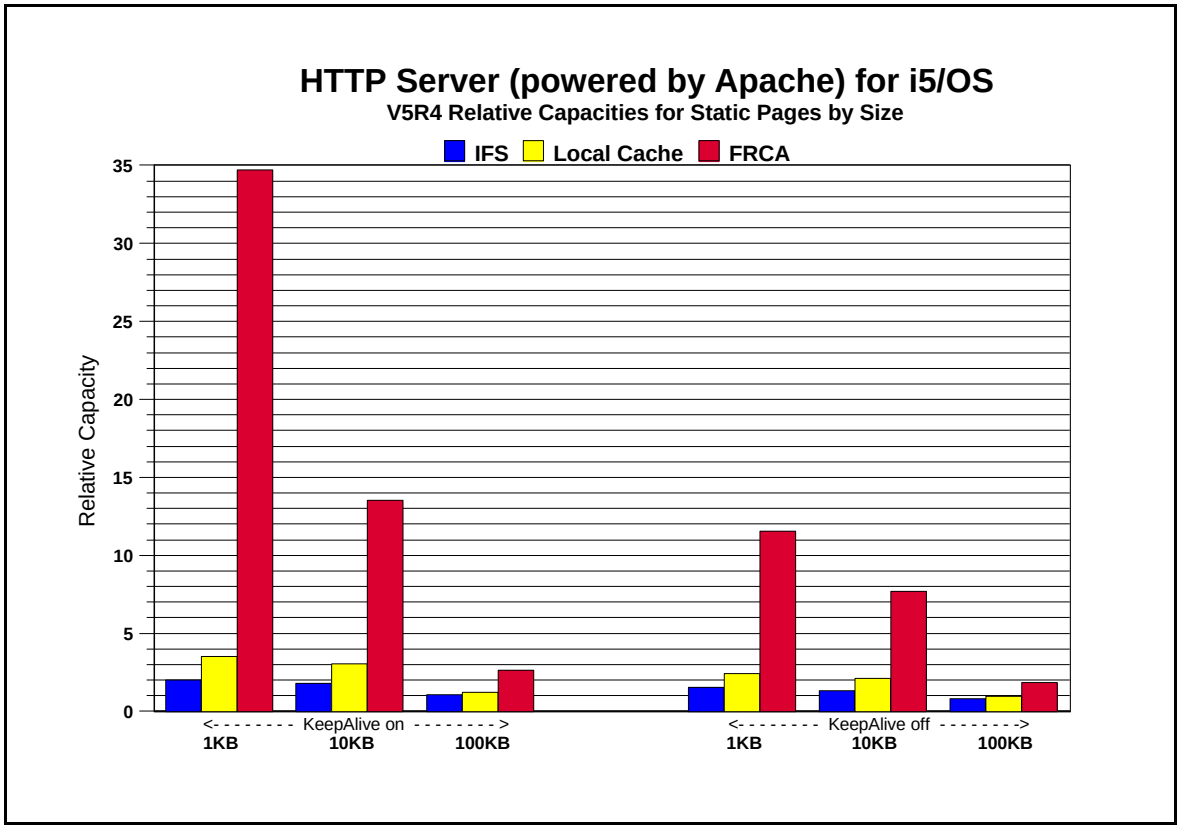


Figure 15.3 i5/OS V5R4 Web Serving Relative Capacity for Static Pages and FRCA

Web Serving Performance Tips and Techniques:

1. **Web Server Cache for IFS Files**: Serving static pages that are cached locally in the HTTP Server's cache can significantly increase Web server capacity (refer to Table 15.3 and Figure 15.3). Ensure that highly used files are selected to be in the cache to limit the overhead of accessing IFS. To keep the cache most useful, it may be best not to consume the cache with extremely large files. Ensure that highly used small/medium files are cached. Also, consider using the LiveLocalCache off directive if possible. If the files you are caching do not change, you can avoid the processing associated with checking each file for any updates to the data. A great deal of caution is recommeded before enabling this directive**.**

2. **FRCA:** Fast Response Caching Accelerator is newly implemented for V5R2. FRCA is based on AFPA (Adaptive Fast Path Architecture), utilizes NFC (Network File Cache) to cache files, and interacts closely with the HTTP Server (powered by Apache). FRCA greatly improves Web server performance for serving static content (refer to Table 15.3 and Figure 15.3). For best performance, FRCA should be used to store static, non-secure content (pages, gifs, images, thumbnails). Keep in mind that HTTP requests served by FRCA are not authenticated and that the files served by FRCA need to have an ASCII CCSID and correct authority. Taking advantage of all levels of caching is really the key for good e-Commerce performance (local HTTP cache, FRCA cache, WebSphere Commerce cache, etc.).

3. **Page size**: The data in the Table 15.1 and Table 15.2 assumes that a small amount of data is being served (say 100 bytes). Table 15.3 illustrates the impact of serving larger files. If the pages are larger, more bytes are processed, CPU processing per transaction significantly increases, and therefore the transaction capacity metrics are reduced. This also increases the communication throughput, which can be a limiting factor for the larger files. The IBM Systems Workload Estimator can be used for capacity planning with page size variations (see chapter 20).

4. **CGI with named activations**: Significant performance benefits can be realized by compiling a CGI program into a "named" versus a "new" activation group, perhaps up to 5x better. It is essential for good performance that CGI-based applications use named activation groups. Refer to the i5/OS ILE Concepts for more details on activation groups. When changing architectures, recompiling CGI programs could boost server performance by taking advantage of compiler optimizations.

5. **Secure Web Serving**: Secure Web serving involves additional overhead to the server for Web environments. There are primarily two groups of overhead: First, there is the fixed overhead of establishing/closing a secure connection, which is dominated by key processing. Second, there is the variable overhead of encryption/decryption, which is proportional to the number of bytes in the transaction. Note the capacity factors in the tables above comparing non-secure and secure serving. From Table 15.1, note that simple transactions (e.g., static page serving), the impact of secure serving is around 20%. For complex transactions (e.g., CGI, servlets), the overhead is more watered down. This relationship assumes that KeepAlive is used, and therefore the overhead of key processing can be minimized. If KeepAlive is not used (i.e., a new connection, a new cached or abbreviated handshake, more key processing, etc.), then there will be a hit of 7x or more CPU time for using secure transaction. To illustrate this, a noncached SSL static transaction using KeepAlive has a relative capacity of 1.481(from Table 15.1); this compares to 0.188 (not included in the table) when KeepAlive is off. However, if the handshake is forced to be a regular or full handshake, then the CPU time hit will be around 50x (relative capacity 0.03). The lesson here is to: 1) limit the use of security to where it is needed, and 2) use KeepAlive if possible.

6. **Persistent Requests and KeepAlive**: Keeping the TCP/IP connection active during a series of transactions is called persistent connection. Taking advantage of the persistent connection for a series of Web transactions is called Persistent Requests or KeepAlive. This is tuned to satisfy an entire typical Web page being able to serve all imbedded files on that same connection.

    a. Performance Advantages: The CPU and network overhead of establishing and closing a connection is very significant, especially for secure transactions. Utilizing the same connection for several transactions usually allows for significantly better performance, in terms of reduced resource consumption, higher potential capacity, and lower response time.

    b. The down side: If persistent requests are used, the Web server thread associated with that series of requests is tied up (only if the Web Server directive AsyncIO is turned Off). If there is a shortage of available threads, some clients may wait for a thread non-proportionally long. A time-out parameter is used to enforce a maximum amount of time that the connection and thread can remain active.

7. **Logging:** Logging (e.g., access logging) consumes additional CPU and disk resources. Typically, it may consume 10% additional CPU. For best performance, turn off unnecessary logging.

8. **Proxy Servers:** Proxy servers can be used to cache highly-used files. This is a great performance advantage to the HTTP server (the originating server) by reducing the number of requests that it must serve. In this case, an HTTP server would typically be front-ended by one or more proxy servers. If the file is resident in the proxy cache and has not expired, it is served by the proxy server, and the back-end HTTP server is not impacted at all. If the file is not cached or if it has expired, then a request is made to the HTTP server, and served by the proxy.

9. **Response Time (general):** User response time is made up of Web browser (client work station) time, network time, and server time. A problem in any one of these areas may cause a significant performance problem for an end-user. To an end-user, it may seem apparent that any performance problem would be attributable to the server, even though the problem may lie elsewhere. It is common for pages that are being served to have imbedded files (e.g., gifs, images, buttons). Each of these transactions may be a separate Internet transaction. Each adds to the response time since they are treated as independent HTTP requests and can be retrieved from various servers (some browsers can retrieve multiple URLs concurrently). Using Persistent Connection or KeepAlive directive can improve this.

10. **HTTP and TCP/IP Configuration Tips:** Information to assist with the configuration for TCP/IP and HTTP can be viewed at http://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp and http://www.ibm.com/servers/eserver/iseries/software/http/

    a. The **number of HTTP server threads**: The reason for having multiple server threads is that when one server is waiting for a disk or communications I/O to complete, a different server job can process another user's request. Also, if persistent requests are being used and AsyncIO is Off, a server thread is allocated to that user for the entire length of the connection. For N-way systems, each CPU may simultaneously process server jobs. The system will adjust the number of servers that are needed automatically (within the bounds of the minimum and maximum parameters). The values specified are for the number of "worker" threads. Typically, the default values will provide the best performance for most systems. For larger systems, the maximum number of server threads may have to be increased. A starting point for the maximum number of threads can be the CPW value (the portion that is being used for Web server activity) divided by

20. Try not to have excessively more than what is needed as this may cause unnecessary system activity.

b. The **maximum frame size parameter** (MAXFRAME on LIND) is generally satisfactory for Ethernet because the default value is equal to the maximum value (1.5K). For Token-Ring, it can be increased from 1994 bytes to its maximum of 16393 to allow for larger transmissions.

c. The **maximum transmission unit (MTU) size** parameter (CFGTCP command) for both the route and interface affect the actual size of the line flows. Optimizing the MTU value will most likely reduce the overall number of transmissions, and therefore, increase the potential capacity of the CPU and the IOP. The MTU on the interface should be set to the frame size (*LIND). The MTU on the route should be set to the interface (*IFC). Similar parameters also exist on the Web browsers. The negotiated value will be the minimum of the server and browser (and perhaps any bridges/routers), so increase them all.

d. Increasing the **TCP/IP buffer size** (TCPRCVBUF and TCPSNDBUF on the CHGTCPA or CFGTCP command) from 8K bytes to 64K bytes (or as high as 8MB) may increase the performance when sending larger amounts of data. If most of the files being served are 10K bytes or less, it is recommended that the buffer size is not increased to the max of 8MB because it may cause a negative effect on throughput.

e. **Error and Access Logging**: Having logging turned on causes a small amount of system overhead (CPU time, extra I/O). Typically, it may increase the CPU load by 5-10%. Turn logging off for best capacity. Use the Administration GUI to make changes to the type and amount of logging needed.

f. **Name Server Accesses**: For each Internet transaction, the server accesses the name server for information (IP address and name translations). These accesses cause significant overhead (CPU time, comm I/O) and greatly reduce system capacity. These accesses can be eliminated by editing the server's config file and adding the line: "HostNameLookups Off".

11. **HTTP Server Memory Requirements**: Follow the faulting threshold guidelines suggested in the work management guide by observing/adjusting the memory in both the machine pool and the pool that the HTTP servers run in (WRKSYSSTS). Factors that may significantly affect the memory requirements include using larger document sizes and using CGI programs**.**

12. **File System Considerations**: Web serving performance varies significantly based on which file system is used. Each file system has different overheads and performance characteristics. Note that serving from the ROOT or QOPENSYS directories provide the best system capacity. If Web page development is done from another directory, consider copying the data to a higher-performing file system for production use. The Web serving performance of the non-thread-safe file systems is significantly less than the root directory. Using QDLS or QSYS may decrease capacity by 2-5 times. Also, be sensitive to the number of sub-directories. Additional overhead is introduced with each sub-directory you add due to the authorization checking that is performed. The HTTP Server serves the pages in ASCII, so make sure that the files have the correct format, else the HTTP Server needs to convert the pages which will result in additional overhead.

13. **Communications/LAN IOPs**: Since there are a dozen or more line flows per transaction (assuming KeepAlive is off), the Web serving environment utilizes the IOP more than other communications environments. Use the Performance Monitor or Collection Services to measure IOP utilization.

Attempt to keep the average IOP utilization at 60% or less for best performance. IOP capacity depends on page size, the MTU size, the use of KeepAlive directive, etc. For the best projection of IOP capacity, consider a measurement and observe the IOP utilization.

15.2 PHP - Zend Core for i

This section discusses the different performance aspects of running PHP transaction based applications using Zend Core for i, including DB access considerations, utilization of RPG program call, and the benefits of using Zend Platform.

Zend Core for i

Zend Core for i delivers a rapid development and production PHP foundation for applications using PHP running on i with IBM DB2 for i or MySQL databases. Zend Core for i includes the capability for Web servers to communicate with DB2 and MySQL databases. It is easy to install, and is bundled with Apache 2, PHP 5, and PHP extensions such as ibm_db2.

The PHP application used for this study is a DVD store application that simulates users logging into an online catalog, browsing the catalog, and making DVD purchases. The entire system configuration is a two-tier model with tier one executing the driver that emulates the activities of Web users. Tier two comprises the Web application server that intercepts the requests and sends database transactions to a DB2 for i or MySQL server, configured on the same machine.

System Configuration

The hardware setup used for this study comprised a driver machine, and a separate system that hosted both the web and database server. The driver machine emulated Web users of an online DVD store generating HTTP requests. These HTTP requests were routed to the Web server that contained the DVD store application logic. The Web server processed the HTTP requests from the Web browsers and maintained persistent connections to the database server jobs. This allowed the connection handle to be preserved after the transaction completed; future incoming transactions re-use the same connection handle. The web and database server was a 2 processor partition on an IBM System i Model 9406-570 server (POWER5 2.2 Ghz) with 2GB of storage. Both IBM i 5.4 and 6.1 were used in the measurements, but for this workload there was minimal difference between the two versions.

Database and Workload Description

The workload used simulates an Online Transaction Processing (OLTP) environment. A driver simulates users logging in and browsing the catalog of available products via simple search queries. Returning customers are presented with their online purchase transactions history, while new users may register to create customer accounts. Users may select items they would like to purchase and proceed to check out or continue to view available products. In this workload, the browse-buy ratio is 5:1. In total, for a given order (business transaction) there are 10 web requests consisting of login, initiate shopping, five product browse requests, shopping cart update, checkout, and product purchase. This is a transaction oriented workload, utilizing commit processing to insure data integrity. In up to 2% of the orders, rollbacks occur due to insufficient product quantities. Restocking is done once every 30 seconds to replenish the product quantities to control the number of rollbacks.

Performance Characterization

The metrics used to characterize the performance of the workload were the following:

- Throughput - Orders Per Minute (OPM). Each order actually consists of 10 web requests to complete the order.
- Order response time (RT) in milliseconds
- Total CPU - Total system processor utilization
- CPU Zend/AP - CPU for the Zend Core / Apache component.
- CPU DB - CPU for the DB component

Database Access

The following four methods were used to access the backend database for the DVD Store application. In the first three cases, SQL requests were issued directly from the PHP pages. In the fourth case, the i5 PHP API toolkit program call interface was used to call RPG programs to issue i5 native DB IO. For all the environments, the same presentation logic was used.

- ibm_db2 extension shipped with Zend Core for i that provides the SQL interface to DB2 for i.
- mysqli extension that provides the SQL interface to MySQL databases. In this case the MySQL InnoDB and MyISAM storage engines were used.
- i5 PHP API Toolkit SQL functions included with Zend Core for i that provide an SQL interface to DB2 for i.
- i5 PHP API Toolkit classes included with Zend Core for i that provide a program call interface.

When using ibm_db2, there are two ways to connect to DB2. If empty strings are passed for userid and password on the connect, the database access occurs within the same job that the PHP script is executing in. If a specific userid and password are used, database access occurs via a QSQSRVR job, which is called server mode processing. In all tests using ibm_db2, server mode processing was used. This may have a minimal performance impact due to management of QSQSRVR jobs, but does prevent the apache job servicing the php request from not responding if a DB error occurs.

When using ibm_db2 and the i5 toolkit (SQL functions), the accepted practice of using prepare and execute was utilized. In addition stored procedures were utilized for processing the purchase transactions. For MySQL, prepared statements were not utilized because of performance overhead.

Finally, in the case of the i5 PHP API toolkit and ibm_db2, persistent connections were used. Persistent connections provides dramatic performance gains versus using non-persistent connections. This is discussed in more detail in the next section.

In the following table, we compare the performance of the different DB access methods.

OS / DB	i 5.4 / DB2	i 5.4 / MySQL 5.0	i 5.4 / DB2	i 5.4 / DB2
ZendCore Version	V2.5.2	V2.5.2	V2.5.2	V2.5.2
Connect	db2_pconnect	mysqli	i5_pconnect	i5_pconnect
			SQL function	Pgm Call Function
OPM	4997	3935	3920	5240
RT (ms)	176	225	227	169
Total CPU	99	98	99	98
CPU - Zend/AP	62	49	63	88
CPU - DB	33	47	33	7

Conclusions:

1. The performance of each DB connection interface provides exceptional response time at very high throughput. Each order processed consisted of ten web requests. As a result, the capacity ranges from about 650 transactions per second up to about 870 transactions per second. Using Zend Platform will provide even higher performance (refer to the section on Zend Platform).

2. The i5 PHP API Toolkit is networked enabled so provides the capability to run in a 3-tier environment, ie, where the PHP application is running on web server deployed on a separate system from the backend DB server. However, when running in a 2- tier environment, it is recommended to use the ibm_db2 PHP extension to access DB2 locally given the optimized performance.

The i5 PHP API Toolkit provides a wealth of interfaces to integrate PHP pages with native i5 system services. When standardizing on the use of the i5 toolkit API, the use of the SQL functions to access DB2 will provide very good performance. In addition to SQL functions, the toolkit provides a program call interface to call existing programs. Calling existing programs using native DB IO may provide significantly more performance.

3. The most compelling reason to use MySQL on IBM i is when you are deploying an application that is written to the MySQL database.

Database - Persistent versus Non-Persistent Connections

If you're connecting to a DB2 database in your PHP application, you'll find that there are two alternative connections - db2_connect which establishes a new connection each time and db2_pconnect which uses persistent connections. The main advantage of using a persistent connection is that it avoids much of the initialization and teardown normally associated with getting a connection to the database. When db2_close() is called against a persistent connection, the call always returns TRUE, but the underlying DB2 client connection remains open and waiting to serve the next matching db2_pconnect() request.

One main area of concern with persistent connections is in the area of commitment control. You need to be very diligent when using persistent connections for transactions that require the use of commitment control boundaries. In this case, DB2_AUTOCOMMIT_OFF is specified and the programmer controls the commit points using db2_commit() statements. If not managed correctly, mixing managed commitment control and persistent connections can result in unknown transaction states if errors occur.

In the following table, we compare the performance of utilizing non-persistent connections in all cases versus using a mix of persistent and non-persistent connections versus using persistent connections in all cases.

OS / DB	i 5.4 / DB2	i 5.4 / DB2	i 5.4 / DB2
ZendCore Version	V2.5.2	V2.5.2	V2.5.2
Connect	db2_connect	Mixed	db2_pconnect
OPM	445	2161	4997
RT (ms)	2021	414	176
Total CPU	91	99	99
CPU - Zend/AP	9	33	62
CPU - DB	78	62	33

Conclusions:

1. As stated earlier, persistent connections can dramatically improve overall performance. When using persistent connections for all transactions, the DB CPU utilization is significantly less than when using non-persistent connections.

2. For any transactions that run with autocommit turned on, use persistent connections. If the transaction requires that autocommit be turned off, use of non-persistent connections may be sufficient for pages that don't have heavy usage. However, if a page is heavily used, use of persistent connections may be required to achieve acceptable performance. In this case, you will need a well designed transaction that handles error processing to ensure no commits are left outstanding.

Database - Isolation Levels

Because the transaction isolation level determines how data is locked and isolated from other processes while the data is being accessed, you should select an isolation level that balances the requirements of concurrency and data integrity. DB2_I5_TXN_SERIALIZABLE is the most restrictive and protected transaction isolation level, and it incurs significant overhead. Many workloads do not require this level of isolation protection. We did limited testing comparing the performance of using DB2_I5_TXN_READ_COMMITTED versus DB2_I5_TXN_READ_UNCOMMITTED isolation levels. With this workload, running under DB2_I5_TXN_READ_COMMITTED reduced the overall capacity by about 5%. However a given application might never update the underlying data or run with other concurrent updaters and DB2_I5_TXN_READ_UNCOMMITTED may be sufficient. Therefore, review your isolation level requirements and adjust them appropriately.

Zend Platform

Zend Platform for i is the production environment that ensures PHP applications are always available, fast, reliable and scalable on the i platform. Zend Platform provides caching and optimization of compiled PHP code, which provides significant performance improvement and scalability. Other features of Zend Platform that brings additional value, include:

- 5020 Bridge – API for accessing 5250 data streams which allows Web front ends to be created for existing applications.
- PHP Intelligence – provides monitoring of PHP applications and captures all the information needed to pinpoint the root cause of problems and performance bottlenecks.
- Online debugging and immediate error resolution with Zend Studio for i
- PHP/Java integration bridge

By automatically caching and optimizing the compiled PHP code, application response time and system capacity improves dramatically. The best part for this is that no changes are required to the take advantage of this optimization. In the measurements included below, the default Zend Platform settings were used.

OS / DB	i 6.1 / DB2		i 6.1/MySQL 5.0	
Zend Version	V2.5.2	V2.5.2/Platform	V2.5.2	V2.5.2/Platform
Connect	db2_pconnect	db2_pconnect	mysqli	mysqli
OPM	5041	6795	3974	4610
RT (ms)	176	129	224	191
Total CPU	98	95	98	96
CPU - Zend/AP	62	44	49	31
CPU - DB	31	46	47	62

Conclusions:

1. In both cases above, the overall system capacity improved significantly when using Zend Platform, by about 15-35% for this workload. With each order consisting of 10 web requests, processing 6795 orders per minute translates into about 1132 transactions per second.

2. Zend Platform will reduce the amount of processing in the Zend Core component since the PHP code is compiled once and reused. In both of the above cases, the amount of processing done in Zend Core on a per transaction basis was dramatically reduced by a factor of about 1.9X.

PHP System Sizing

The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE) is a web-based sizing tool for IBM Power Systems, System i, System p, and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. The Estimator allows measurement input to best reflect your current workload and provides a variety of built-in workloads to reflect your emerging application requirements.

Currently, a new built-in workload is being developed to allow the sizing of PHP workloads on Power Systems running IBM i. This built-in is expected to be available November 2008. To access WLE use the following URL:

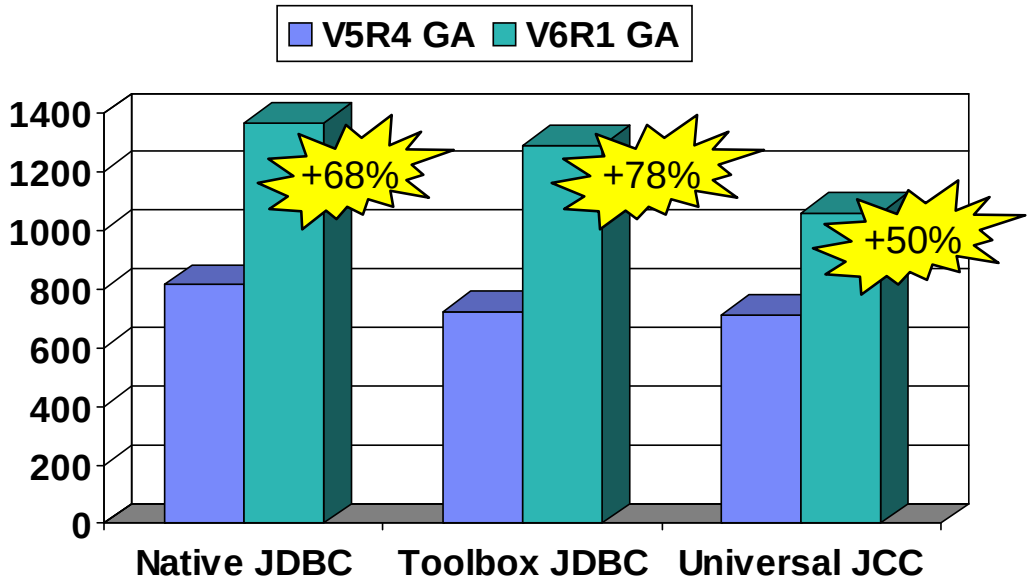
http://www.ibm.com/eserver/iseries/support/estimator

15.3 WebSphere Application Server

This section discusses System i performance information for the WebSphere Application Server, including WebSphere Application Server V6.1, WebSphere Application Server V6.0, WebSphere Application Server V5.0 and V5.1, and WebSphere Application Server Express V5.1. Historically, both WebSphere and i5/OS Java performance improve with each version. Note from the figures and data in this section that the most recent versions of WebSphere and/or i5/OS generally provides the best performance.

What's new in V6R1?

The release of i5/OS V6R1 brings with it significant performance benefits for many WebSphere applications. The following chart shows the amount of improvement in transactions per second (TPS) for the Trade 6.1 workload using various data access methods:



This chart shows that in V6R1, throughput levels for Trade 6.1 increased from 50% to nearly 80% versus V5R4, depending on which JDBC provider was being used. All measurement results were obtained in a 2-tier environment (both application and database on the same partition) on a 2-core 2.2Ghz System i partition, using Version 6.1 of WebSphere Application Server and IBM Technology for Java VM. Although many of the improvements are applicable to 3-tier environments as well, the communications overhead in these environments may affect the amount of improvement that you will see.

The improvements in V6R1 were primarily in the JDBC, DB2 for i5/OS and Java areas, as well as changes in other i5/OS components such as seize/release and PASE call overhead. The majority of the improvements will be achieved without any changes to your application, although some improvements do require additional tuning (discussed below in **Tuning Changes for V6R1**). Although some of the changes are now available via V5R4 PTFs, the majority of the improvement will only be realized by moving to V6R1. The actual amount of improvement in any particular application will vary, particularly depending on the amount of JDBC/DB activity, where a significant majority of the changes were made. In addition,

because the improvements largely resulted from significant reductions in pathlength and CPU, environments that are constrained by other resources such as IO or memory may not show the same level of improvements seen here.

Tuning changes in V6R1

As indicated above, most improvements will require no changes to an application. However, there are a few changes that will require some tuning in order to be realized:

- **Using direct map (native JDBC)**

  For System i, the JDBC interfaces run more efficiently if direct mapping of data is used, where the data being retrieved is in a form that closely matches how the data is stored in the database files. In V6R1, significant enhancements were made in JDBC to allow direct map to be used in more cases. For the toolbox and JCC JDBC drivers, where direct map is the default, there is no change needed to realize these gains. However, for native JDBC, you will need to use the "directMap=true" custom property for the datasource in order to maximize the gain from these changes. For Trade 6.1, measurements show that adding this property results in about a 3-5% improvement in throughput. Note that there is no detrimental effect from using this property, since the JDBC interfaces will only use direct map if it is functionally viable to do so.

- **Use of unix sockets (toolbox JDBC)**

  For toolbox JDBC, the default is to use TCP/IP inet sockets for requests between the application server and the database connections. In V6R1, an enhancement was added to allow the use of unix sockets in a 2-tier toolbox environment (application and database reside on the same partition). Using unix sockets for the Trade 6.1 2-tier workload in V6R1 resulted in about an 8-10% improvement in throughput. However, as the default is still to use inet sockets, you will need to ensure that the class path specified in the JDBC provider is set to use the jt400native.jar file (not the jt400.jar file) in order to use unix sockets. Note that the improvement is applicable only to 2-tier toolbox environments. Inet sockets will continue to be used for all other multiple tier toolbox environments no matter which .jar file is used.
  .

- **Using 'threadUsed=false" custom property (toolbox JDBC)**

  In toolbox JDBC, the default method of operation is to use multiple application server threads for each request to a database connection, with one thread used for sending data to the connection and another thread being used to receive data from the connection. In V6R1, changes were made to allow both the send and receive activity to be done within a single application server thread for each request, thus reducing the overhead associated with the multiple threads. To gain the potential improvement from this change, you will need to specify the "threadUsed=false" custom property in the toolbox datasource, since the default is still to use multiple threads. For the Trade 6.1 workload, use of the this property resulted in about a 10% improvement in throughput.

Tuning for WebSphere is important to achieve optimal performance. Please refer to the *WebSphere Application Server for iSeries Performance Considerations or the WebSphere Info Center* documents for more information. These documents describe the performance differences between the different WebSphere Application Server versions on the IBM i platform. They also contain many performance recommendations for environments using servlets, Java Server Pages (JSPs), and Enterprise Java Beans.

For WebSphere 5.1, 6.0 and 6.1 please refer to the following page and follow the appropriate link:
www.ibm.com/software/webservers/appserv/was/library/

For WebSphere 7.0 please refer to InfoCenter:
http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp

For tuning specific information on WebSphere 7.0 visit this link:
WebSphere 7.0 performance tuning for IBM i

Although some capacity planning information is included in these documents, please use the IBM Systems Workload Estimator as the primary tool to size WebSphere environments. The Workload Estimator is kept up to date with the latest capacity planning information available.

Trade 6 Benchmark (IBM Trade Performance Benchmark Sample for WebSphere Application Server**)**
Description:

Trade 6 is the fourth generation of the WebSphere end-to-end benchmark and performance sample application. The Trade benchmark is designed and developed to cover the significantly expanding programming model and performance technologies associated with WebSphere Application Server. This application provides a real-world workload, enabling performance research and verification test of the Java™ 2 Platform, Enterprise Edition (J2EE™) 1.4 implementation in WebSphere Application Server, including key performance components and features**.**

Overall, the Trade application is primarily used for performance research on a wide range of software components and platforms. This latest revision of Trade builds off of Trade 3, by moving from the J2EE 1.3 programming model to the J2EE 1.4 model that is supported by WebSphere Application Server V6.0. Trade 6 adds `DistributedMap` based data caching in addition to the command bean caching that is used in Trade 3. Otherwise, the implementation and workflow of the Trade application remains unchanged.

Trade 6 also supports the recent DB2® V8.2 and Oracle® 10g databases. The new design of Trade 6 enables performance research on J2EE 1.4 including the new Enterprise JavaBeansTM (EJB™) 2.1 component architecture, message-driven beans, transactions (1-phase, 2-phase commit) and Web services (SOAP, WSDL, JAX-RPC, enterprise Web services). Trade 6 also drives key WebSphere Application Server performance components such as dynamic caching, WebSphere Edge Server, and EJB caching.

NOTE: Trade 6 is an updated version of Trade 3 which takes advantage of the new JMS messaging support available with WebSphere 6.0. The application itself is essentially the same as Trade 3 so direct comparisons can be made between Trade 6 and Trade 3. However, it is important to note that direct comparisons between Trade2 and Trade3 are NOT valid. As a result of the redesign and additional components that were added to Trade 3, Trade 3 is more complex and is a heavier application than the previous Trade 2 versions.

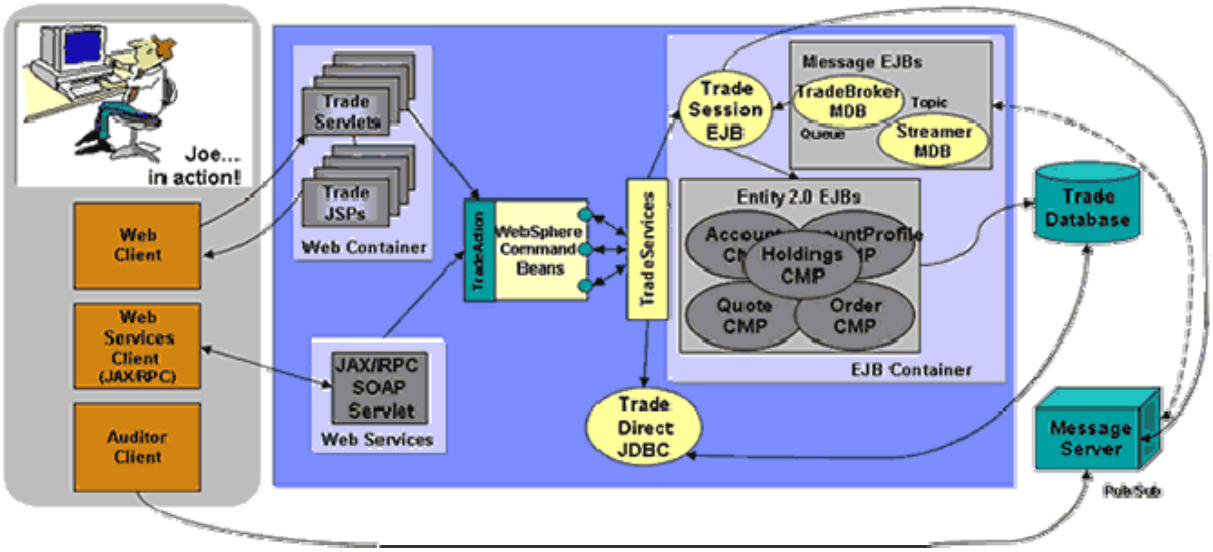


Figure 15. 1 Topology of the Trade Application

The Trade 6 application allows a user, typically using a Web browser, to perform the following actions:

- Register to create a user profile, user ID/password and initial account balance
- Login to validate an already registered user
- Browse current stock price for a ticker symbol
- Purchase shares
- Sell shares from holdings
- Browse portfolio
- Logout to terminate the users active interval

Each **action** is comprised of many primitive operations running within the context of a single HTTP request/response. For any given action there is exactly one transaction comprised of 2-5 remote method calls. A ***Sell*** action for example, would involve the following primitive operations:

- Browser issues an HTTP GET command on the TradeAppServlet
- TradeServlet accesses the cookie-based HTTP Session for that user
- HTML form data input is accessed to select the stock to sell
- The stock is sold by invoking the **sell()** method on the **Trade** bean, a stateless ***Session EJB***. To achieve the sell, a transaction is opened and the Trade bean then calls methods on Quote, Account and Holdings ***Entity EJBs*** to execute the sell as a single transaction.
- The results of the transaction, including the new current balance, total sell price and other data, are formatted as HTML output using a Java Server Page, portfolio.jsp.
- Message Driven Beans are used to inform the user that the transaction has completed on the next logon of that user.

To measure performance across various configuration options, the Trade 6 application can be run in several modes. A mode defines the environment and component used in a test and is configured by modifying settings through the Trade 6 interface. For example, data object access can be configured to use JDBC directly or to use EJBs under WebSphere by setting the Trade 6 *runtime mode*. In the ***Sell*** example above, operations are listed for the EJB runtime mode. If the mode is set to JDBC, the *sell* action is completed by direct data access through JDBC from the TradeAppServlet. Several testing modes are available and are varied for individual tests to analyze performance characteristics under various configurations.

WebSphere Application Server V6.1

Historically, new releases of WebSphere Application Server have offered improved performance and functionality over prior releases of WebSphere. WebSphere Application Server V6.1 is no exception. Furthermore, the availability of WebSphere Application Server V6.1 offers an entirely new opportunity for WebSphere customers. Applications running on V6.1 can now operate with either the "Classic" 64-bit Virtual Machine (VM) or the recently released IBM Technology for Java, a 32-bit VM that is built on technology being introduced across all the IBM Systems platforms.

Customers running releases of WebSphere Application prior to V6.1 will likely be familiar with the Classic 64-bit VM. This continues to be the default VM on i5/OS, offering competitive performance and excellent vertical scalability. Experiments conducted using the Trade6 benchmark show that WebSphere Application Server V6.1 running on the Classic VM realized performance gains of 5-10% better throughput when compared to WebSphere Application Server V6.0 on identical hardware.

In addition to the presence of the Classic 64-bit VM, WebSphere Application Server V6.1 can also take advantage of IBM Technology for Java, a 32-bit implementation of Java supported on Java 5.0 (JDK 1.5). For V6.1 users, IBM Technology for Java has two key potentially beneficial characteristics:

- *Significant performance improvements for many applications* - Most applications will see at least equivalent performance when comparing WebSphere Application Server on the Classic VM to IBM Technology for Java, with many applications seeing improvements of up to 20%.
- *32-bit addressing allows for a potentially considerable reduction in memory footprint* - Object references require only 4 bytes of memory as opposed to the 8 bytes required in the 64-bit Classic VM. For users running on small systems with relatively low memory demands this could offer a substantially smaller memory footprint. Performance tests have shown approximately 40% smaller Java Heap sizes when using IBM Technology for Java when compared to the Classic VM.

It is important to realize that both the Classic VM and IBM Technology for Java have excellent benefits for different applications. Therefore, choosing which VM to use is an extremely important consideration. *Chapter 14 - Java Performance* has an extensive overview of many of the key decisions that go into choosing which VM to use for a given application. Most of the points in Chapter 14 are very much important to WebSphere Application Server users. One issue that will likely not be a concern to WebSphere Application Server users is the additional overhead to native ILE calls that is seen in IBM Technology for Java. However, if native calls are relevant to a particular application, that consideration will of course be important. While choosing the appropriate VM is important, WebSphere Application Server V6.1 allows users to toggle between the Classic VM and IBM Technology for Java either for the entire WebSphere installation or for individual application server profiles.

While 32-bit addressing can provide smaller memory footprints for some applications, it is imperative to understand the other end of the spectrum: applications requiring large Java heaps may not be able to fit in the space available to a 32-bit implementation of Java. The 32-bit VM has a maximum heap size of 3328 MB for Java applications. However, WebSphere Application Server V6.1 using IBM Technology for Java has a practical maximum heap size of around 2500 MB due in part to WebSphere related memory demands like shared classes. The Classic VM should be used for applications that require a heap larger than 2500 MB (see Chapter 14 - Java Performance for further details).

Trade Measurement Results:

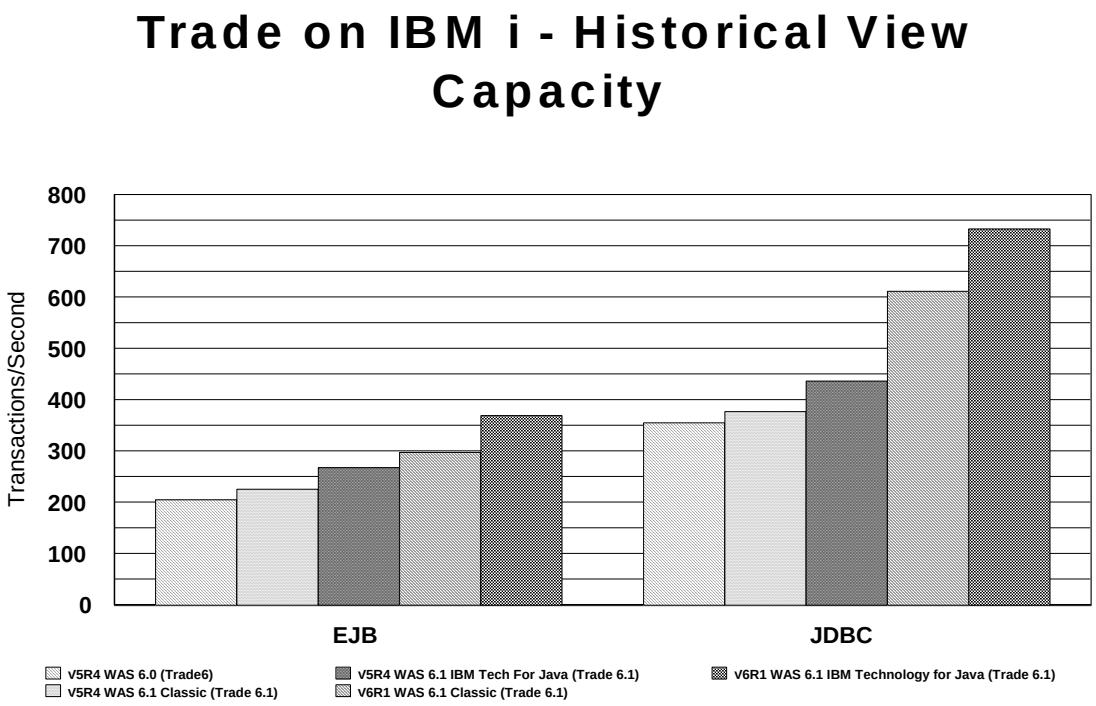


Figure 15.2 Trade Capacity Results

WebSphere Application Server Trade Results
Notes/Disclaimers:
• Trade chart: WebSphere 6.0 was measured on V5R4 on a 2 way (LPAR) 570/7758 system WebSphere 6.1 using Classic VM (V5R4) was measured on a 2 way (LPAR) 570/7758 system WebSphere 6.1 using IBM Technology for Java (V5R4) was measured on a 2 way (LPAR) 570/7758 system WebSphere 6.1 using Classic VM (V6R1) was measured on a 2 way (LPAR) 570/7758 system WebSphere 6.1 using IBM Technology for Java (V6R1) was measured on a 2 way (LPAR) 570/7758 system

Trade Scalability Results:

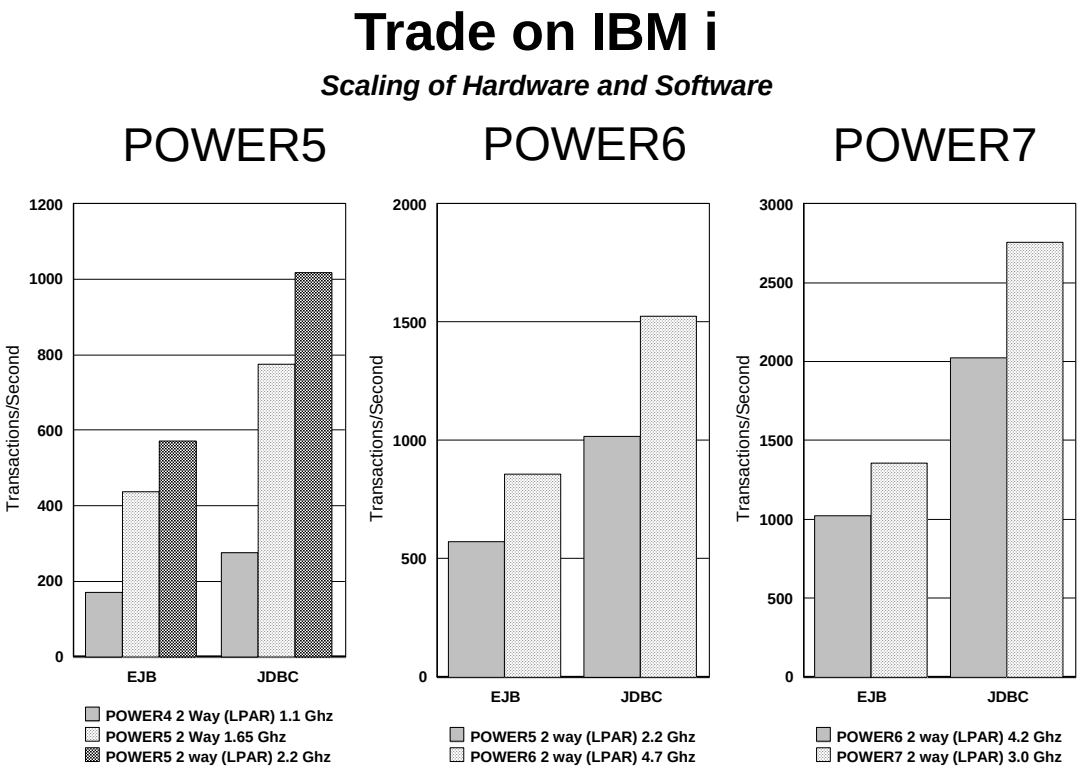


Figure 15.3 Trade Scaling Results

WebSphere Application Server Trade Results
Notes/Disclaimers:
• POWER5 chart: POWER4 - V5R3 825/2473 2-Way (LPAR) 1.1 GHz., POWER4 was measured with WebSphere 5.1 POWER5 - V5R3 520/7457 2-Way 1.65 GHz., POWER5 was measured with WebSphere 5.1 POWER5 - V5R4 570/7758 2-Way (LPAR) 2.2 GHz, POWER5 was measured with WebSphere 6.0 • POWER6 chart: POWER5 - V5R4 570/7758 2-Way (LPAR) 2.2 GHz, POWER5 was measured with WebSphere 6.0 POWER6 - V5R4 9406-MMA 2-Way (LPAR) 4.7 GHz, POWER6 was measured with WebSphere 6.1 • POWER7 chart: POWER6 - V6R1 Model 570 (9117-MMA) 2-Way (LPAR) 4.2 GHz, POWER6 was measured with WebSphere 7.0 POWER7 - V6R1 Model 750 (8233-E8B) 2-Way (LPAR) 3.0 GHz, POWER7 was measured with WebSphere 7.0 NOTE: Throughput comparisons should not be made between charts

Accelerator for System i

Coinciding with the release of i5/OS V5R4, IBM introduced entry IBM System i models. The models introduce accelerator technologies and/or L3 cache in order to improve options for clients in the low-end server space. As an overview, the Accelerator for System i affects two 520 Models: (1) 600 CPW with no L3 cache and (2) 1200 CPW with L3 cache. With the Accelerator for System i, the 600 CPW can be accelerated to a 3100 CPW system, whereas the 1200 CPW can be accelerated to 3800 CPW.
In order to showcase the abilities of these systems, experiments were completed on WAS 6.0 running Trade 6 to display the benefits. The following information describes the models in the context of both capacity and response time. Results were collected on System i Model 520 with varying feature codes depending on the presence of the Accelerator for System i.
With regards to capacity, Figure 15.5 shows the 600 CPW model accelerated to 3100 CPW increases capacity 5.5 times. Additionally, the 1200 CPW model accelerated to 3800 CPW increases capacity 3 times. This provides an extraordinary benefit when running WebSphere Applications.

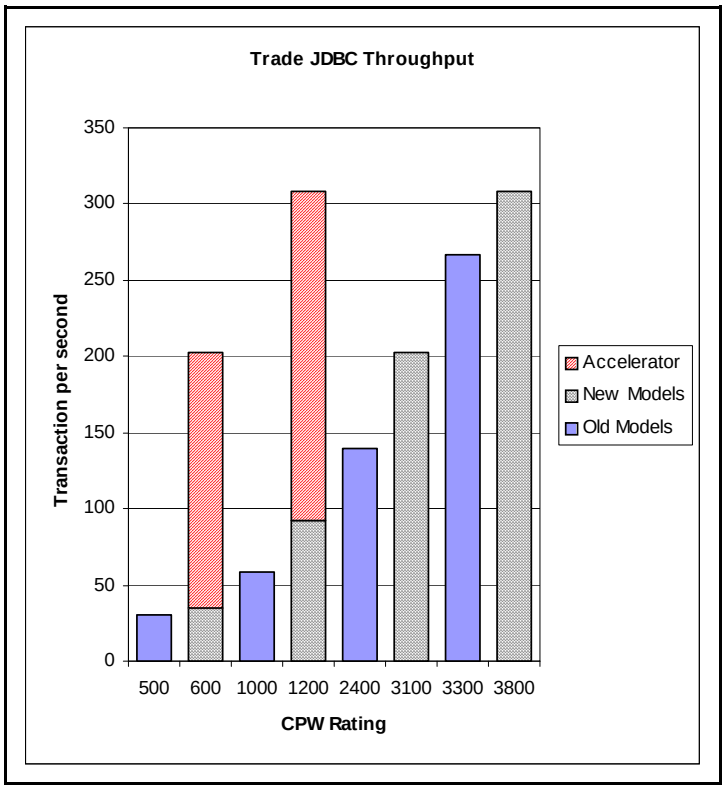


Figure 15.5 - Accelerator for System i performance data - Capacity comparison (WAS 6.0 running Trade 6).

It is also important to note the benefits of L3 cache. For example, the 1200 CPW model has 2.5 times more capacity than that of the 600 CPW system. Additionally, Java workloads tend to perform better with L3 cache. Thus, besides the benefit of increased capacity, a move from a system with no L3 cache to a system with L3 cache may scale better than CPW ratings would indicate.

Figure 15.6 provides insight into response time information regarding low-end System i models. There are two key concepts that are displayed in the data in Figure 15.6. The first is that Accelerator for System i models can provide substantially better response times than previous models for a single or many users. The 600 CPW accelerated to 3100 CPW reduces the response time by 5 times while the 1200 CPW accelerated to 3800 CPW reduces the response time by 2.5 times. The second idea to note is that the presence of L3 cache has little effect on the response time of a single user. Of course there are benefits of L3 cache, however, the absence of L3 cache does not imply poorer performance with regards to response time.

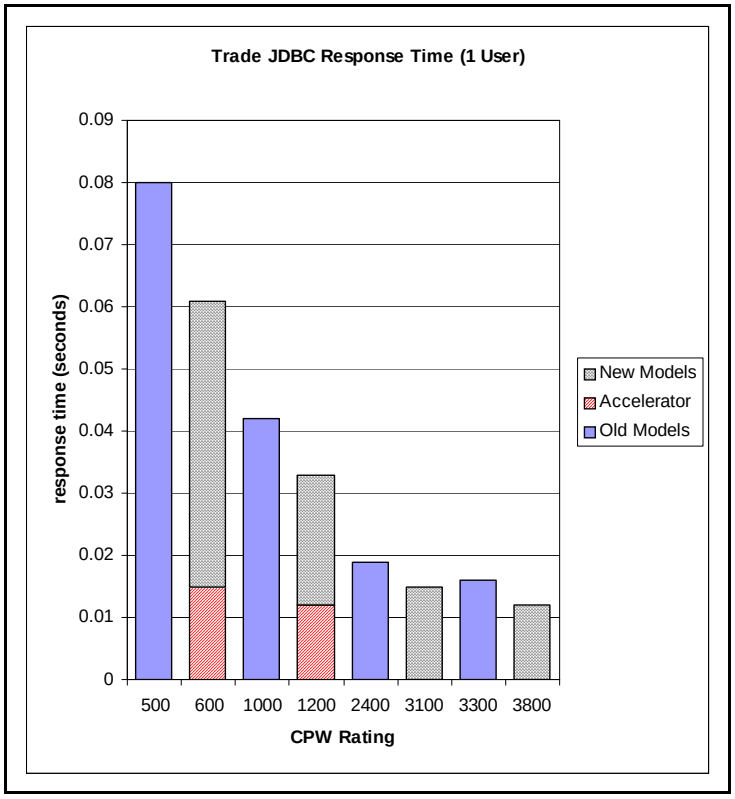


Figure 15.6 - Accelerator for System i performance data - Single user response time comparison (WAS 6.0 running Trade 6).

Performance Considerations When Using WebSphere Transaction Processing (XA)

Generally, a transaction is the execution of a set of related operations that must be completed together. This set of operations is referred to as a unit-of-work. A transaction is said to commit when it completes successfully, otherwise it is said to roll back. When an application needs to access more than one resource and needs to guarantee job consistency, a global transaction is required to coordinate the interactions among the resources, application and transaction manager, as defined by XA specification.

WebSphere Application Server is compliant with the XA specification. It uses the two-phase commit protocol to guarantee that either all the resources commit the change permanently or none of them precede the update (rollback). Such a transaction scenario requires the resource managers, such as WebSphere JMS/MQ and DB2 UDB, to support the XA specification and provide the XA interface for a two-phase commit. It is the role of the resource managers to manage access to shared resources involved in a transaction and guarantee that the ACID properties (Atomicity, Consistency, Isolation, and Durability) of a transaction are maintained. It is the role of WebSphere Transaction manager to control all of the global transaction logic as defined by the J2EE Standard. Within WebSphere there are two ways of using WebSphere global transaction: Container Managed Transaction (CMT) and Bean Managed Transaction (BMT). With Container Managed, the J2EE container, WebSphere in this case, controls all the transaction logic.

When your application involves multiple resources, such as DB2, in a transaction, you need to ensure that you select an XA compliant JDBC provider. For WebSphere on the System i platform you have two options depending on if you are running in a two tier environment (application server and database server on the same system) or in a three tier environment (application server and database server on separate systems). However, since the overhead of running XA is quite significant, you should ensure that you do not configure an XA compliant JDBC provider if your application does not require XA functionality.

In WebSphere 6.0 the JMS provider was totally rewritten. It is now 100% pure Java and it no longer requires WebSphere MQ to be installed. As a result, you can configure your application to share the JDBC connection used by a messaging engine, and the EJB container. This enables you to use one-phase commit (non-XA) transactions since there is only one resource manager (DB2) involved in a transaction. By utilizing one-phase commit optimization, you can improve the performance of your application.

You can benefit from the one-phase commit optimization in the following circumstances:
- Your application must use the assured persistent reliability attribute for its JMS messages.
- Your application must use CMP entity beans that are bound to the same JDBC data source that the messaging engine uses for its data store.

Restriction: You cannot benefit from the one-phase commit optimization in the following circumstances:
- If your application uses a reliability attribute other than assured persistent for its JMS messages.
- If your application uses Bean Managed Persistence (BMP) entity beans, or JDBC clients.

Before you configure your system, ensure that you consider all of the components of your J2EE application that might be affected by one-phase commits. Also, since the JDBC datasource connection will now be shared by the messaging engine and the EJB container, you need to ensure that you increase the number of connections allocated to the connection pool. To optimize for one-phase commit transactions, refer to the following website:
Http://publib.boulder.ibm.com/infocenter/ws60help/index.jsp?topic=/com.ibm.websphere.pmc.doc/tasks/tjm0280_.html

15.4 IBM WebFacing

The IBM WebFacing tool converts your 5250 application DDS display files, menu source, and help files into Java Servlets, JSPs, JavaBeans, and JavaScript to allow your application to run in either WebSphere Application Server V5 or V4. This is an easy way to bring your application to either the Internet, or the Intranet, both quickly and inexpensively.

The Number of Screens processed per second and the number of Input/Output fields per screen are the main metric to tell how heavy a WebFaced application will be on the WebSphere Application Server. The number of Input/Output fields are simple to count for most of the screens, except when dealing with subfiles. Subfiles can affect the number of input/output fields dramatically. The number of fields in subfiles are significantly impacted by two DDS keywords:

1. SFLPAG - The number of rows shown on a 5250 display.
2. SFLSIZ - The number of rows of data extracted from the database.

When using a DDS subfile, there are 3 typical modes of operation:

1. SFLPAG=SFLSIZ. In this mode, there are no records that are cached. When more records are requested, WebFacing will have to get more rows of data. This is the recommended way to run your WebFacing application.
2. SFLPAG < SFLSIZ. In this mode, WebFacing will get SFLSIZ rows of data at a time. WebFacing will display SFLPAG rows, and cache the rest of the rows. When the user requests more rows with a page-down, WebFacing will not have to access the database again, unless they page below the value of SFLSIZ. When this happens, WebFacing will go back to the database and receive more rows.
3. SFLPAG = (SFLSIZ) * (Number of times requesting the data). This is a special case of option 2 above, and is the recommended approach to run GreenScreen applications. For the first time the page is requested, SFLPAG rows will be returned. If the user performs a page down, then SFLPAG * 2 rows will be returned. This is very efficient in 5250 applications, but less efficient with WebFacing.

Since WebFacing is performance sensitive to the number of input/output fields that are requested from WebFacing, the best option would be the first mode, since this will minimize the number of these fields for each 5250 panel requested through WebFacing. The number of fields for a subfile is the number of rows requested from the database (SFLSIZE) times the number of columns in each row.

Figure 15.7 shows a theoretical algorithm to graphically describe the effect the number of Input/Output fields has on the performance of the WebFaced application. The Y-axis metric is not important, but merely can be used to measure the relative amount of CPU horsepower that the application needs to serve one single 5250 panel. In this case, serving one single panel with 50 I/O fields is approximately one half the CPU horsepower needed to serve one 5250 panel with 350 I/O fields. As you can see, the number of I/O fields dramatically impacts the performance of your WebFacing application, thereby reducing the I/O fields will improve your performance.

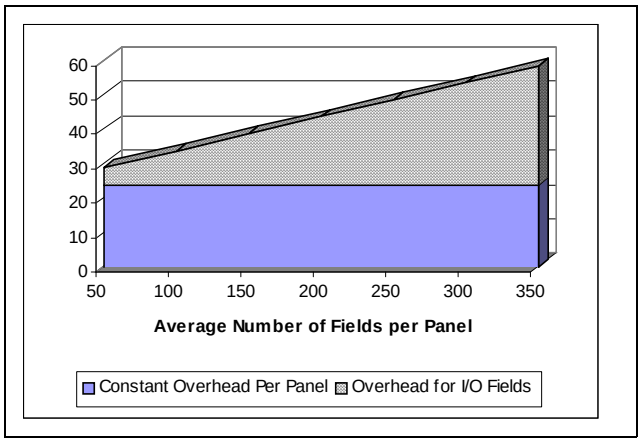


Figure 15.7 Shows the impact on CPU that the number of I/O fields has per WebFaced panel

In our studies, we selected three customer WebFaced applications, one simple, one moderate, and one complex. See table 15.4, for

details on the number of I/O fields for each of these workloads. We ran the workloads on three separate machines (see table 15.5) to validate the performance characteristics with regard to CPW. In our running of the workloads, we tolerated only a 1.5 second **server response time** per panel. This value does not include the time it takes to render the image on the client system, but only the time it took the server to get the information to the client system. The machines that we used are in Table 15.5, and include the 800 and i810 (V5R2 Hardware) and the 170 (V4R4 Hardware). All systems were running OS/400 V5R2.

Some of the results that we saw in our tests are shown in Figure 15.8. This figure shows the scalability across different hardware running the same workload. A user is defined as a client that requests one new 5250 panel every 15 seconds. According to our tests, we see relatively even results across the three machines. The one machine that is a slight difference is the V4R4 hardware (1090 CPW). This slight difference can be explained by release-to-release degradation. Since the CPW measurement were made in V4R4, there have been three major releases, each bringing a slight degradation in performance. This results in a slight difference in CPW value. With this taken into effect, the CPW/User measurement is more in line with the other two machines.

Name	Average number of I/O Fields / panel
Workload A	37
Workload B	99
Workload C	612

Table 15.4 Average number of I/O fields for each workload defined in this section.

Many 5250 applications have been implemented with "best performance" techniques, such as minimized number of fields and amount of data exchanged between the device and the application. Other 5250 applications may not be as efficiently implemented, such as restoring a complete window of data, when it was not required. Therefore it is difficult to give a generalized performance comparison between the same application written to a 5250 device and that application using WebFacing to a browser. In the three workloads that we measured, we saw a significant amount of resource needed to WebFace these applications. The numbers varied from 3x up to 8x the amount of CPU resources needed for the 5250 green screen application.

Use the IBM Systems Workload Estimator to predict the capacity characteristics for IBM WebFacing This site will be updated, more often then this paper, so it will contain the most recent information. The Workload Estimator will ask you to specify a transaction rate (5250 panels per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of the panels and the number of unique panels that are displayed by the JSP. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (also see chapter 20).

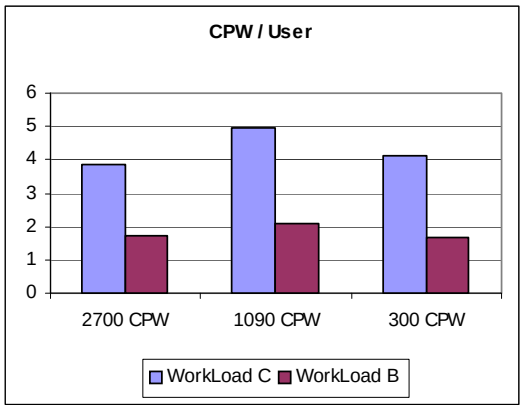


Figure 15.8 CPW per User across the machines documented in table 15.5

Display File Record I/O Processing

Display file record I/O processing has been optimized to decrease the WebSphere Application Server runtime memory utilization. This has been accomplished by enhancing the Webfacing runtime to better utilize the java objects required for processing display I/O requests for each end user transaction. Formerly on each record I/O, Webfacing had to create a record

data bean object to describe the I/O request, and then create the record bean using this definition to pass the I/O data to the associated JSP. These definition objects were not reused and were created for each user. With the optimization implemented in V5.0, the record bean definitions are now reused and cached so that one instance for each display file record can be shared by all users.

This optimization has decreased the overall memory requirements for Webfacing V5.0 versus V4.0. This memory savings helps reduce the total memory required by the WebSphere Application Server, which is referred to as the JVM Heap Size. The amount of memory savings depends on a number of parameters, such as the complexity of the screens (based on number of fields per screen), the transaction rate, and the number of concurrent end users. On measurements made with approximately 250 users and varying screen complexity, the JVM Heap decreased by approximately 5 % for simple to moderate screens (99 fields per screen) and up to 20 % for applications with more complex screens (600 fields per screen). When looking at the overall memory requirements for an application, the JVM Heap size is just one component. If you are running the back-end application on the same server as the WebSphere Application server, the overall decrease in system memory required for the Webfaced application will be less.

In terms of WebSphere CPU utilization, this optimization offers up to a 10% improvement for complex workloads. However, when taking into account the overall CPU utilization for a Webfaced application (Webfacing plus the application), you can expect equal or slightly better performance with Webfacing V5.0.

Tuning the Record Definition Cache

In order to best use the optimization provided by this enhancement, servlet utilities have been included in the Webfacing support to assess cache efficiency, set the cache size, and preload it with the most frequently accessed record definitions. If you do not use the Record Definition Cache, or it is not tuned properly, you will see degraded performance of Webfacing V5.0 versus V4.0.

When set to an appropriate level for the Webfaced application, the Record Definition Cache can provide a decrease in memory usage, and slightly decreased processor usage. The number of record definitions that the cache will retain is set by an initialization parameter in the Webfaced application's deployment descriptor (web.xml). By changing the cache size, the Webfaced application can be tuned for best performance and minimum memory requirements. The cache size determines the number of record data definitions that will be retained in the cache. There is one record data definition for each record format.

Cache Size	Effect
too small	When the cache size is set too small for the Webfaced application it will adversely affect the performance. In this case, the definitions would be cached then discarded before being re-used. There is significant overhead to create the record definitions.
correct	With the cache set correctly, 90% of all accessed record data definitions would be retained in the cache with few cache misses for not commonly used records.
too large	If the cache is set too large then all record data definitions for the Webfaced application would be cached, likely consuming memory for seldom used definitions.

In order to determine what is the correct size for a given Webfaced application, the number of commonly used record formats needs to estimated. This can be used as a starting point for setting the cache size.

The default size, if no size is specified, would be 600 record data definitions. To set the cache size to something other than the default size, you need to add a session context parameter in the Webfaced application's web.xml file. In the following example the cache size is set to 200 elements, which may be appropriate for a very small application, like the Order Entry example program.

```
<context-param>
  <param-name>WFBeanCacheSize</param-name>
  <param-value>200</param-value>
  <description>WebFacing Record Definition Bean Cache Size</description>
</context-param>
```

NOTE: For information on defining a session context parameter in the web.xml file, refer to the WebSphere Application Server Info Center. You can also edit the web.xml file of a deployed application. Typically this file will be located in the following directory for WebSphere V5.0 applications:

/QIBM/UserData/WebAS5/Base/<application-server>/config/cells/..../WEB_INF

And the following directory for WebSphere Express V5.0 applications:

/QIBM/UserData/WebASE/ASE5/<application-server>/config/cells/..../WEB_INF

Cache Management - Definition Cache Content Viewer

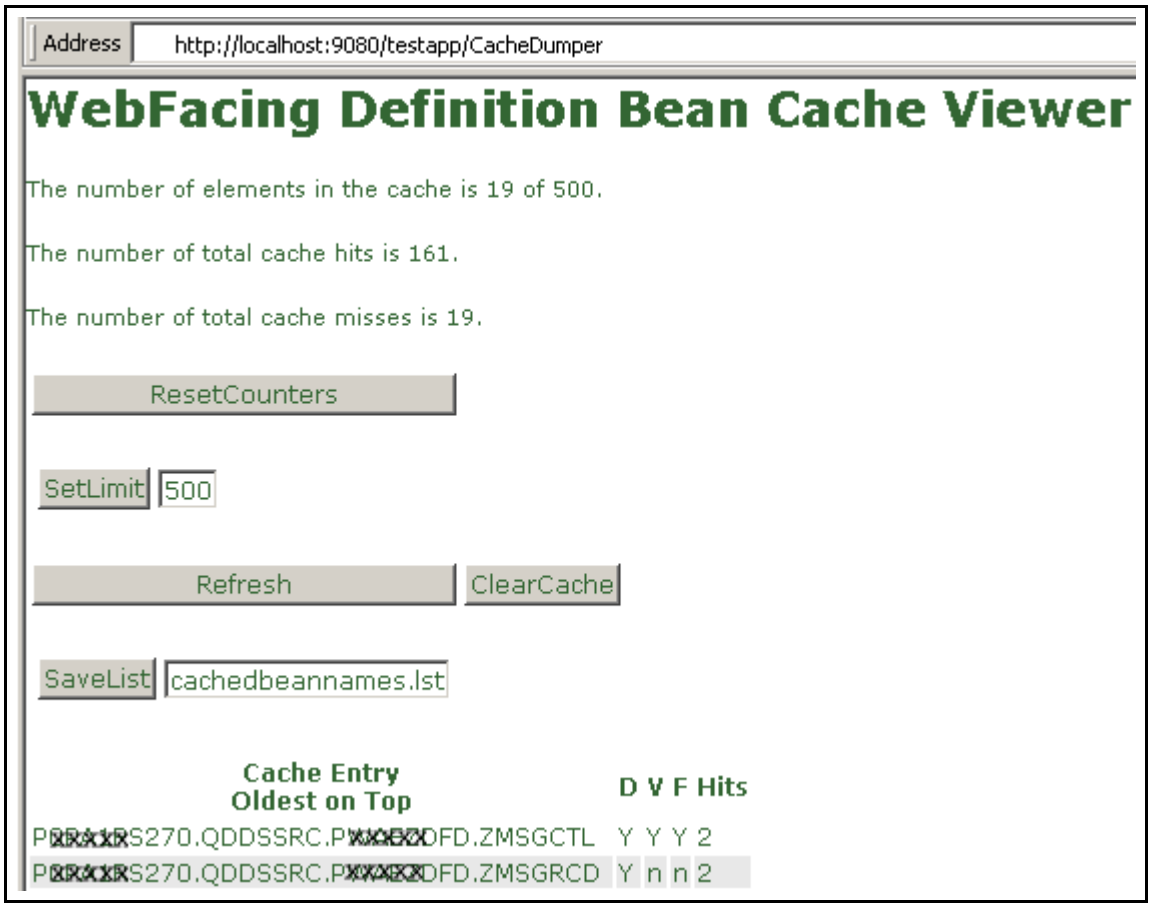
To assist with managing the Record Definition Cache, two servlets can be enabled. One is used to display the elements currently in the cache and the other can be used to load the cache. Both of these servlets are not normally enabled in a WebFacing application in order to prevent mis-use or exposure of data.
To enable the servlet that will display the contents of the cache, first add the following segments to the Webfaced application's web.xml.

```
<servlet>
  <servlet-name>CacheDumper</servlet-name>
  <display-name>CacheDumper</display-name>
  <servlet-class>com.ibm.etools.iseries.webfacing.diags.CacheDumper</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>CacheDumper</servlet-name>
  <url-pattern>/CacheDumper</url-pattern>
</servlet-mapping>
```

This servlet can then be invoked with a URL like: http://<server>:<port>/<webapp>/CacheDumper.

Then a Web page like that shown below will be displayed. Notice that the total number of cache hits and misses are displayed, as are the hits for each record definition.



Refer to the following table for the functionality provided by the Cache Viewer servlet.

Cache Viewer Button operations

Button	Operation
Reset Counters	Resets the cache hit and miss counters back to 0.
Set Limit	Temporarily sets the cache limit to a new value. Setting the value lower than the current value will cause the cache to be cleared as well.
Refresh	Refresh the display of cache elements.
Clear Cache	Drop all the cached definitions.
Save List	Save a list of all the cached record data definitions. This list is saved in the RecordJSPs directory of the Webfaced application. The actual record definitions are not saved, just the list of what record definitions are cached. Once the cache is optimally tuned, this list can be used to preload the Record Definition cache.

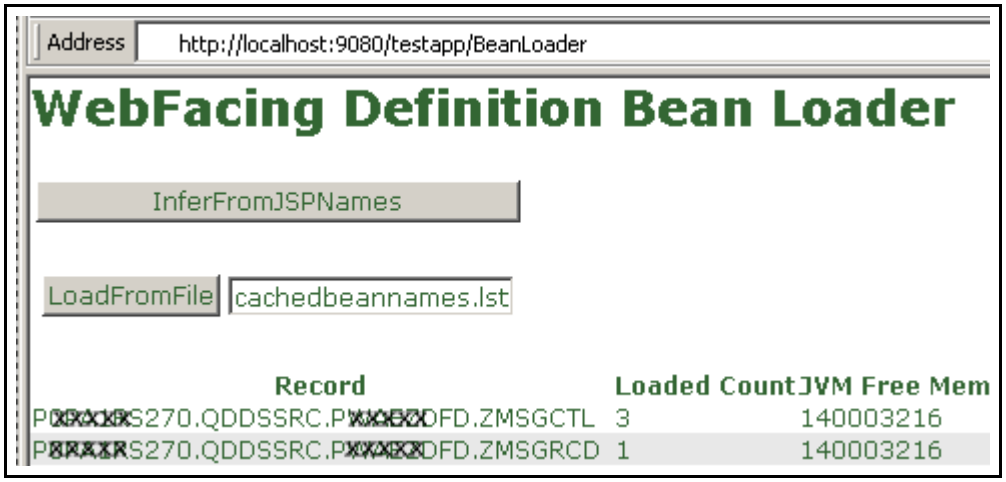
Cache Management - Record Definition Loader

As a companion to the Cache Content Viewer tool, there is also a Record Definition Cache Loader tool, which is also referred to as the Bean Loader. This servlet can be used to pre-load the cache to aid in the determination of the optimal cache size, and then finally, to pre-load the cache for production use. To enable this servlet add the following two xml segments in the web.xml file.

```
<servlet>
  <servlet-name>BeanLoader</servlet-name>
  <display-name>BeanLoader</display-name>
  <servlet-class>com.ibm.etools.iseries.webfacing.diags.BeanLoader</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>BeanLoader</servlet-name>
  <url-pattern>/BeanLoader</url-pattern>
</servlet-mapping>
```

Invoking this servlet will present a Web page similar to the following.



Refer to the following table for the functionality provided by the Record Definition Loader servlet.

Record Definition Loader Button operations

Button	Operation
Infer from JSP Names	This will cause the loader servlet to infer record definition names from the names or the JSP's contained in the RecordJsps directory. It will not find all the record definitions but it will get most of them.
Load from File	This option will load the record definitions listed in a file in the RecordJSPs directory. Typically this file is created with the CacheDumper servlet previously described.

The Record Definition Loader servlet can also be used to pre-load the bean definitions when the Webfaced application is started. To enable this the servlet definition in the web.xml needs to be updated to define two init parameters: FileName and DisableUI. The FileName parameter indicates the name of the file in the RecordJSPs directory that contains the list of definitions to pre-load the cache with. The

DisableUI parameter indicates that the Web UI (as presented above) would be disabled so that the servlet can be used to safely pre-load the definitions without exposing the Webfaced application.

```
<servlet>
 <servlet-name>BeanLoader</servlet-name>
 <display-name>BeanLoader</display-name>
 <servlet-class>com.ibm.etools.iseries.webfacing.diags.BeanLoader</servlet-class>
 <init-param>
  <param-name>FileName</param-name>
  <param-value>cachedbeannames.lst</param-value>
 </init-param>
 <init-param>
  <param-name>DisableUI</param-name>
  <param-value>true</param-value>
 </init-param>
 <load-on-startup>10</load-on-startup>
 </servlet>
```

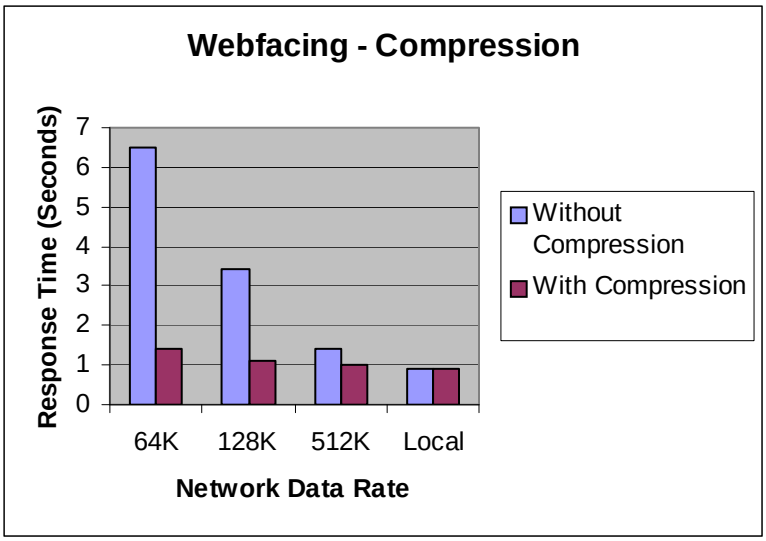
Compression

LAN connection speeds and Internet hops can have a large impact on page response times. A fast server but slow LAN connection will yield slow end-user performance and an unhappy customer.

It is very common for a browser page to contain 15-75K of data. Customers who may be running a Webfaced application over a 256K internet connection might find results unacceptable. If every screen averages 60K, the time for that data spent on the wire is significant. Multiply that by several users simultaneously using the application, and page response times will be longer.

There are now two options available to support HTTP compression for Webfaced applications, which will significantly improve response times over a slow internet connection. As of July 1, 2003, compression support was added with the latest set of PTFs for IBM HTTP Server (powered by Apache) for i5/OS (5722-DG1). Also, Version 5.0 of Webfacing was updated to support compression available in WebSphere Application Server. On System i servers, the recommended WebSphere application configuration is to run Apache as the web server and WebSphere Application Server as the application server. Therefore, it is recommended that you configure HTTP compression support in Apache. However, in certain instances HTTP compression configuration may be necessary using the Webfacing/WebSphere Application Server support. This is discussed below.

The overall performance in both cases is essentially equivalent. Both provide significant improvement for end-user response times on slower Internet connections, but also require additional HTTP/WebSphere Application Server CPU resources. In measurements done with compression, the amount of CPU required by HTTP/WebSphere Application Server increased by approximately 25-30%. When compression is enabled, ensure that there is sufficient CPU to support it. Compression is particularly beneficial when end users are attached via a Wide Area Network (WAN) where the network connection speed is 256K or less. In these cases, the end user will realize significantly improved response times (see chart below). If the end users are attached via a 512K connection, evaluate whether the realized response time improvements offset the increased CPU requirements. Compression should not be used if end users are connected via a local intranet due to the increased CPU requirements and no measurable improvement in response time.



NOTE: The above results were achieved in a controlled environment and may not be repeatable in other environments. Improvements depend on many factors.

Enabling Compression in IBM HTTP Server (powered by Apache)

The HTTP compression support was added with the latest set of PTFs for IBM HTTP Server for i5/OS (5722-DG1). For V5R1, the PTFs are SI09287 and SI09223. For V5R2, the PTFs are SI09286 and SI09224.

There is a LoadModule directive that needs to be added to the HTTP config file in order to get compression based on this new support. It looks like this:

LoadModule deflate_module /QSYS.LIB/QHTTPSVR.LIB/QZSRCORE.SRVPGM

You also need to add the directive:

SetOutputFilter DEFLATE

to the container to be compressed, or globally if the compression can always be done. There is documentation on the Apache website on mod_deflate (http://httpd.apache.org/docs-2.0/mod/mod_deflate.html) that has information specific to setting up for compression. That is the best place to look for details. The LoadModule and SetOutputFilter directives are required for mod_deflate to work. Any other directives are used to further define how the compression is done.

Since the compression support in Apache for i5/OS is a recent enhancement , Information Center documentation for the HTTP compression support was not available when this paper was created. The IBM HTTP Server or i5/OS website (http://www.ibm.com/servers/eserver/iseries/software/http/) will be updated with a splash when the InfoCenter documentation has been completed. Until the documentation is available, the information at http://httpd.apache.org/docs-2.0/mod/mod_deflate.html can be used as a reference for tuning how mod_deflate compression is done.

Enabling Compression using IBM Webfacing Tool and WebSphere Application Server Support

You would configure compression using the Webfacing/WebSphere support in environments where the internal HTTP server in WebSphere Application Server is used. This may be the case in a test environment, or in environments running WebSphere Express V5.0 on an xSeries Server.

With the IBM WebFacing Tool V5.0, compression is 'turned on' by default. This should be 'turned off' if compression is configured in Apache or if the LAN environment is a local high speed connection. This is particularly important if the CPU utilization of interactive types of users (Priority 20 jobs) is about 70-80% of the interactive capacity. In order to 'turn off' compression, edit the web.xml file for a deployed Web application. There is a filter definition and filter mapping definition that defines compression should be used by the WebFacing application (see below). These statements should be deleted in order to 'turn off' compression. In a future service pack of the WebFacing Tool, it is planned that compression will be configurable from within WebSphere Development Studio Client.

```
<filter id="Filter_1051910189313">
        <filter-name>CompressionFilter</filter-name>
        <display-name>CompressionFilter</display-name>
        <description>WebFacing Compression Filter</description>
        <filter-class>com.ibm.etools.iseries.webfacing.runtime.filters.CompressionFilter</filter-class>
 </filter>
     <filter-mapping id="FilterMapping_1051910189315">
        <filter-name>CompressionFilter</filter-name>
        <url-pattern>/WFScreenBuilder</url-pattern>
 </filter-mapping>
```

Additional Resources

The following are additional resources that include performance information for Webfacing including how to setup pretouch support to improve JSP first-touch performance:

PartnerWorld for Developers Webfacing website:
http://www.ibm.com/servers/enable/site/ebiz/webfacing/index.html

IBM WebFacing Tool Performance Update - This white paper expains how to help optimize WebFaced Applications on IBM System i servers. Requests for the paper require user registration; there are no charges.
http://www-919.ibm.com/servers/eserver/iseries/developer/ebiz/documents/webfacing/

15.5 WebSphere Host Access Transformation Services (HATS)

WebSphere Host Access Transformation Services (HATS) gives you all the tools you need to quickly and easily extend your legacy applications to business partners, customers, and employees. HATS makes your 5250 applications available as HTML through the most popular Web browsers, while converting your host screens to a Web look and feel. With HATS it is easy to improve the workflow and navigation of your host applications without any access or modification to source code.

What's new with V5R4 and HATS 6.0.4

The IBM WebFacing Tool has been delivering reliable and customizable Web-enabled applications for years. Host Access Transformation Services (HATS) has been providing seamless runtime Web-enablement. Now, with the IBM WebFacing Deployment Tool with HATS Technology (WDHT), IBM offers a single product with the power of both technologies.

This offering replaces HATS for iSeries and HATS for System i model 520. For HATS applications created using HATS Toolkit 6.0.4 and deployed to a V5R4 system, you can now connect to the WebFacing Server and eliminate the Online Transaction Processing charge. Without the OLTP requirement for deploying a HATS application to i5/OS starting with V5R4, the overall cost of HATS solutions is significantly reduced. HATS applications can now be deployed to i5/OS Standard Edition.

With WDHT, WebFacing applications can call non-WebFacing applications and those programs will be dynamically transformed for the Web using HATS technology.

HATS Customization

HATS uses a rules-based engine to dynamically transform 5250 applications to HTML. The process preserves the flow of the application and requires very little technical skill or customization.

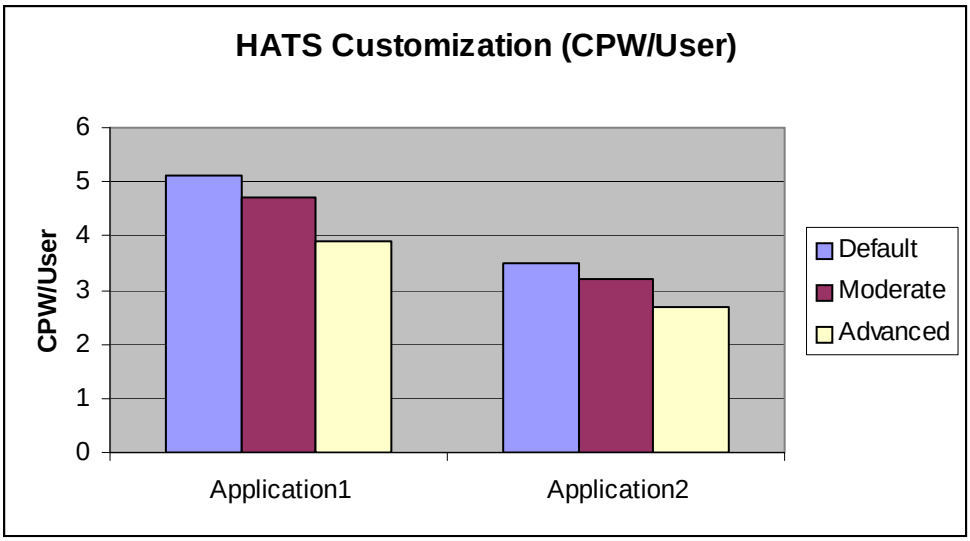
Unless you do explicit customization for an application, the default HATS rules will be used to transform the application interface dynamically at runtime. This is referred to as default rendering. Basically a default template JSP is used for all application screens. There is the capability to change the default template to customize the web appearance, but at runtime the application screens are still dynamically transformed.

As an alternative, you can use HATS studio (built upon the common WebSphere Studio Workbench foundation) to capture and customize select screens or all screens in an application. In this case a JSP is created for each screen that is captured. Then at runtime the first step HATS performs is to check to see if there are any screens that have been captured and identified that match the current host screen. If there are no screen customizations, then the default dynamic transformation is applied. If there is a screen customization that matches the current host screen, then whatever actions have been associated with this screen are executed.

Since default rendering results in dynamic screen transformation at run time, it will require more CPU resources than if the screens of an application have been customized. When an application is customized, JSPs are created so that much of the transformation is static at run time. Based on measurements for a mix of applications using the following levels of customizations, Moderate Customization typically requires 5-10% less CPU as compared to Default Rendering. With Advanced Customization, typically 20-25% less CPU is required as compared to Default Rendering. You have to take into account, though, that

customization requires development effort, while Default Rendering requires minimal development resources.

Default: The screens in the application's main path are unchanged.
Moderate: An average of 30% of the screens have been customized.
Advanced: All screens have been customized.



IBM Systems Workload Estimator for HATS

The purpose of the *IBM Systems Workload Estimator (WLE)* is to provide a comprehensive System i sizing tool for new and existing customers interested in deploying new emerging workloads standalone or in combination with their current workloads. The Estimator recommends the model, processor, interactive feature, memory, and disk resources necessary for a mixed set of workloads. WLE was enhanced to support sizing a System i server to meet your HATS workload requirements.

This tool allows you to input an interactive transaction rate and to further characterize your workload. Refer to the following website to access WLE, http://www.ibm.com/estimator/index.html . Work with your marketing representative to utilize this tool, and also refer to chapter 20 for more information.

15.6 WebSphere Portal

The IBM WebSphere Portal suite of products enables companies to build a portal website serving the individual needs of their employees, business partners and customers. Users can sign on to the portal and view personalized web pages that provide access to the information, people and applications they need. This personalized, single point of access to resources reduces information overload, accelerates productivity and increases website usage. As WebSphere Portal supports access through mobile devices, as well as the desktop browser, critical information is always available. Visit the WebSphere Portal InfoCenter for more information:
http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html

Use the IBM Systems Workload Estimator (Estimator) to predict the capacity characteristics for WebSphere Portal (using the WebSphere Portal workload category). For custom applications, the Workload Estimator will ask you questions about your portal pages served, such as the number of portlets per page and the complexity of each portlet. It will also ask you to specify a transaction rate (visits per hour) for a peak time of day. In addition to custom applications, the Estimator supports Portal Document Manager (PDM) and Web Content Management (WCM) for some releases of WebSphere Portal. Because of potential performance differences between WebSphere Portal releases, the projections for one release cannot be applied to other releases.

- WebSphere Portal 6.0 - Custom applications and PDM.
- WebSphere Portal Express 6.0 - Custom applications, PDM, and WCM.
- WebSphere Portal 6.1 - Custom application and WCM.
- WebSphere Portal 7.0 - Customer application and WCM.

The Estimator is available at: http://www.ibm.com/systems/support/tools/estimator. Extensive descriptions and help text for the Portal workloads are available in the Estimator. Please work with your marketing representative when using the Estimator to size Portal workloads.

15.7 WebSphere Commerce

Use the IBM Systems Workload Estimator to predict the capacity characteristics for WebSphere Commerce performance (using the Web Commerce workload category). The Workload Estimator will

ask you to specify a transaction rate (visits per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of shopping visits (browse/order ratio, number of transactions per user visit, database size, etc.). Recently, the Estimator has also been enhanced to include WebSphere Commerce Pro Entry Edition. The Web Commerce workload also incorporates WebSphere Commerce Payments to process payment transactions. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (see also chapter 20).

To help you tune your WebSphere Commerce website for better performance on the System i platform, there is a performance tuning guide available at: http://www-1.ibm.com/support/docview.wss?uid=swg21198883. This guide provides tips and techniques, as well as recommended settings or adjustments, for several key areas of WebSphere and DB2 that are important to ensuring that your website performs at a satisfactory level.

15.8 WebSphere Commerce Payments

Use the IBM Systems Workload Estimator to predict the capacities and resource requirements for WebSphere Commerce Payments. The Estimator allows you to predict a standalone WCP environment or a WCP environment associated with the buy visits from a WebSphere Commerce estimation. Work with your marketing representative to utilize this tool. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator.

Workload Description: The PayGen workload was measured using clients that emulate the payment transaction initiated when Internet users purchase a product from an e-commerce shopping site. The payment transaction includes the Accept and Approve processing for the initiated payment request. WebSphere Commerce Payments has the flexibility and capability to integrate different types of payment cassettes due to the independent architecture. Payment cassettes are the plugins used to accommodate payment requirements on the Internet for merchants who need to accept multiple payment methods. For more information about the various cassettes, follow the link below: http://www-4.ibm.com/software/webservers/commerce/paymentmanager/lib.html

Performance Tips and Techniques:

1. **DTD Path Considerations:** When using the Java Client API Library (CAL), the performance of the WebSphere Commerce Payments can be significantly improved if the merchant application specifies the dtdPath parameter when creating a PaymentServerClient. When this parameter is specified, the overhead of sending the entire IBMPaymentServer.dtd file with each response is avoided. The dtdPath parameter should contain the path of the locally stored copy of the IBMPaymentServer.dtd file. For the exact location of this file, refer to the *Programmer's Guide and Reference* at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgrprog22as.html

2. **Other Tuning Tips:** More performance tuning tips can be found in the *Administrator's Guide* under Appendix D at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgradmin22as.html

3. **WebSphere Tuning Tips:** Please refer to the WebSphere section in section 15.2, for a discussion on WebSphere Application Server performance as well as related web links.

15.9 WebSphere MQ

The WebSphere MQ allows application programs to communicate with each other using messages and message queuing. The applications can reside either on the same machine or on different machines or platforms that are separated by one or more networks. For example, IBM i applications can communicate with other IBM i applications through WebSphere MQ, or they can communicate with applications on other platforms by using WebSphere MQ and the appropriate MQ Series product(s) for the other platform (HP-UX, Windows, Linux, AIX, etc.).

WebSphere MQ supports all important communications protocols, and shields applications from having to deal with the mechanics of the underlying communications being used. In addition, WMQ ensures that data is not lost due to failures in the underlying system or network infrastructure. Applications can also deliver messages in a time independent mode, which means that the sending and receiving applications are decoupled so the sender can continue processing without having to wait for acknowledgement that the message has been received.

WebSphere MQ 7.0

WebSphere MQ V7.0 on IBM i has similar performance characteristics to the V6 product. Throughput is similar overall (for local, client and distributed queuing) when the clients are running in V6 compatibility mode. The default enhanced client support that provides heartbeating, enhanced reliability and multiplexing degrades client benchmarks by 5-15%. There are new functions in V7 that provide enhanced performance to applications that are able to use them. These include Asynchronous Puts, Read-ahead, Properties, and selectors, but they are not covered in this document.

For further information, please reference the appropriate Performance Evaluations document for WebSphere MQ, found here: http://www.ibm.com/support/docview.wss?rs=171&uid=swg27007150

Other Sources of Information

In addition to the above mentioned support pacs, you can refer to the following URL for reference guides, online manuals, articles, white papers and other sources of information on WebSphere MQs: http://www.ibm.com/software/integration/wmq/

Chapter 16. Lotus Domino on IBM i

Performance information for Lotus Domino on the IBM i operating system can be found in the following articles, redbooks, redpapers, and tools:

- IBM Systems Workload Estimator (WLE):
  http://www.ibm.com/eserver/iseries/support/estimator

- IBM Lotus Domino 8.5 performance for IBM Lotus Notes users, March 2009
  http://www.ibm.com/developerworks/lotus/library/domino85-performance/

- IBM Lotus Domino 8.5 performance for iNotes users, March 2009
  http://www.ibm.com/developerworks/lotus/library/domino85-inotes/

- IBM Lotus Notes V8 workloads: Taking performance to a new level, September 2007
  http://www.ibm.com/developerworks/lotus/library/notes8-workloads/index.html

- IBM Lotus Domino V8 server with the IBM Lotus Notes V8 client: Performance, October 2007
  http://www.ibm.com/developerworks/lotus/library/domino8-performance/index.html

- Lotus Domino 7 Server Performance, Part 1, September 2005
  http://www.ibm.com/developerworks/lotus/library/nd7-perform/index.html

- Lotus Domino 7 Server Performance, Part 2, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-internet-performance/index.html

- Lotus Domino 7 Server Performance, Part 3, November 2005
  http://www.ibm.com/developerworks/lotus/library/domino7-enterprise-performance/

- Best Practices for Large Lotus Notes Mail Files, October 2005
  http://www.ibm.com/developerworks/lotus/library/notes-mail-files/

- Redbooks and Redpapers
  - Implementing IBM Lotus Domino 7 for i5/OS (SG24-7311), April 2007
  - Domino 6 for iSeries Best Practices Guide (SG24-6937), March 2004
  - Lotus Domino 6 for iSeries Multi-Versioning Support on iSeries (SG24-6940), March 2004
  - Sizing Large-Scale Domino Workloads on iSeries (redpaper), December 2003
  - Domino 6 for iSeries Implementation (SG24-6592), February 2003
  - Upgrading to Domino 6: The Performance Benefits (redpaper), January 2003
  - Domino for iSeries Sizing and Performance Tuning (SG24-5162), April 2002
  - iNotes Web Access on the IBM eServer iSeries Server (SG24-6553), February 2002

Chapter 17. Integrated BladeCenter and System x Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

This chapter provides a performance overview and recommendations for iSCSI attached BladeCenter blade and System x servers. In addition, the chapter presents some performance characteristics and impacts of integrated servers on IBM i.

17.1 Introduction

The Internet SCSI (iSCSI) solution extends the utility of IBM i by integrating x86 and AMD based servers with the IBM i platform. The iSCSI solution allows IBM i to integrate and control selected IBM BladeCenter® and System x servers that run Windows® Server 2008 editions, Windows Server 2003 editions, or VMware ESX/ESXi 4 server.

All performance results documented in this chapter were run using integrated Windows servers. VMware ESX was not tested.

For more information about supported models, operating systems, and options, see the "IBM i integration with BladeCenter and System x" web page referenced at the end of this chapter. Also, see the IBM i Information Center content titled "IBM i integration with BladeCenter and System x" for iSCSI concepts and operation details.

In the text following, the IBM i platform is often referred to as the "host" server. The iSCSI attached BladeCenter or System x server is referred to as the "integrated" server.

The IBM i iSCSI solution provides an extensive scalability range - from connecting up to 8 integrated servers through one IBM i iSCSI target for a lower cost connectivity, to allowing up to 4 IBM i iSCSI targets per individual integrated server for scalable bandwidth. For information about the numbers of supported iSCSI adapters, see the "IBM i integration with BladeCenter and System x" web page.

With the iSCSI solution, no disks are installed in the integrated servers. The host IBM i server provides disks, storage consolidation, integrated server management, along with tape, optical, and virtual Ethernet devices.

There are two types of iSCSI initiator and target adapter implementations:

Type	Description
Software initiator or target (Ethernet NIC)	With a **software initiator (SWI)** or a **software target (SWT)**, the iSCSI protocol is implemented in the server operating system. Server resources (for example, CPU and memory) are used for the iSCSI protocol. The IBM i integrated server solution uses standard Ethernet network interface cards (Ethernet NICs) as SWIs and SWTs. Integrated server SWIs and IBM i SWTs support 1 Gbit or 10 Gbit Ethernet network connections, depending on the iSCSI adapters and the network infrastructure that is used.
Hardware initiator or target (iSCSI HBA)	With a **hardware initiator (HWI)** or a **hardware target (HWT)**, the iSCSI protocol is implemented in firmware on the iSCSI adapter. The iSCSI protocol is offloaded from the server.

	The IBM i integrated server solution uses iSCSI host bus adapters (iSCSI HBAs) as HWIs and HWTs. Integrated server HWIs and IBM i HWTs support 1 Gbit Ethernet network connections.

The performance results presented in the rest of this chapter are based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

17.2 Performance tradeoff summary

The following table lists some tradeoffs and considerations for the IBM i iSCSI solution. Some of these items are explained in greater detail in later sections of this chapter.

Tradeoff	Considerations
Software target (SWT) vs. Hardware target (HWT)	Key comparisons: • 1 Gbps SWT vs. 1 Gbps HWT throughput is about the same, except in a 2 initiator to 2 target (2x2) configuration, where SWT outperforms HWT. • 10 Gbps SWT throughput exceeds 1 Gbps HWT throughput by a wide margin (see 10 Gbps vs. 1 Gbps below). • IBM i CPW usage for SWT (with a 512 MB iSCSI memory pool, digests turned off and using jumbo frames) is about 20% higher than with HWT.
Software initiator (SWI) vs. Hardware initiator (HWI)	Throughput is the same, assuming a 1 Gbps connection. However, the integrated server CPU utilization for SWI is higher than for HWI (about 2-2.5 times as much). **Note**: With a Windows server that has many cores and hyper threading, the increase in CPU utilization is hard to notice due to all the processors on the system.

10 Gbps
vs.
1 Gbps
	The 10 Gbps software solution (SWI to SWT) is much faster than 1 Gbps configurations.
	
	Some general notes for 10 Gbps configurations:
	1. Configure jumbo frames for a 10 Gbps SWI to SWT configuration to reduce CPU utilization and to increase throughput.
	2. With regular TCP streaming, the test configuration could not reach the full line speed. The speeds achieved were:
	  • 6-7 Gbps when using jumbo frames
	  • 3-4 Gbps when using standard frames
	3. For the test configuration, the system write throughput was maxed out using jumbo frames (~600 MB/s).
	4. For the test configuration, the system read throughput was maxed out due to the InfiniBand bus (~1000 MB/s).

	5. Since large amounts of data can be transferred with a 10 Gbps configuration, the disk subsystem must be very high performing and the system buses used need ample bandwidth. **Note**: At the time of this writing, the IBM i iSCSI solution does not support 10 Gbps iSCSI connections for VMware ESX/ESXi 4 servers.
512 MB memory pool vs. 10 MB memory pool	A larger shared data memory pool size makes a **big** improvement in throughput and IBM i CPW usage. • For example, one test had a 10 Gb multipath (2x2) SWI to SWT configuration (with digests off and using jumbo frames) and using 4K-32K block sizes. With this configuration, increasing the memory pool size from 10 MB to 512 MB resulted in a **3-5 times improvement** in throughput, while at the same time showing a **45-47% reduction** in IBM i CPW usage. As this example shows, a large memory pool is **very** important. Therefore, the recommended **minimum** memory pool size is **512 MB**. • This memory pool can be shared among all of the iSCSI attached servers on the system. • If excessive page faults occur in the memory pool, increase the memory pool size to see if that helps. **Note**: Prior to IBM i 7.1, the recommended minimum memory pool size was 10 MB when using HWTs. However, HWTs also benefit from using a larger memory pool, although not as dramatic an improvement as the SWT example above. Therefore, 512 MB is now the recommended minimum memory pool size for all configurations, including HWT.
Digests vs. No Digests	iSCSI header and data digests are used to detect errors that occur at the iSCSI layer. When iSCSI digests are disabled, TCP and/or Ethernet error detection mechanisms provide data integrity. Disabling iSCSI digests for a SWT improves throughput some and uses less IBM i CPW. • For example, disabling iSCSI digests for a 1 Gb SWT and using 4K-32K block sizes showed a **7-33% improvement** in IBM i CPW usage. • Disabling iSCSI digests for a 10 Gb SWT improves IBM i CPW usage quite a bit. • Disabling iSCSI digests makes no difference for a HWI or a HWT. **How to turn off digests:** Use of iSCSI digests are configured from the iSCSI initiator. • For HWI to SWT configurations, see the **Disabling iSCSI header and data digests** task in the **iSCSI Initiator Hardware Configuration** document referenced at the end of this chapter. • For SWI to SWT or SWI to HWT configurations, no action is required since SWIs have iSCSI digests disabled by default. • For HWI to HWT configurations, no action is required since disabling iSCSI digests does not provide a performance improvement.
Jumbo frames	For a pure software solution (SWI to SWT), jumbo (9000 byte) frames

vs. Standard frames	improved IBM i CPW usage compared to standard (1500 byte) frames. • For example, using jumbo frames with a 10 Gb SWT and using 4K-32K block sizes showed a **3-21% improvement** in IBM i CPW usage. To enable jumbo frames for SWI to SWT configurations, see the **Changing the iSCSI initiator MTU** task in the **iSCSI Initiator Hardware Configuration** document referenced at the end of this chapter. **Note**: For any configuration involving a HWI or a HWT, performance using jumbo frames is worse than when using standard frames, so use standard frames for these configurations.
Single path I/O vs. Multipath I/O	A 2 initiator to 2 target (2x2) multipath I/O configuration with a round robin policy provided double the throughput of a 1 initiator to 1 target (1x1) configuration.
Ethernet NIC adapter vs. Embedded Ethernet port	For SWT usage, an Ethernet NIC adapter card performs better (less IBM i CPU used) than an Ethernet port that is embedded in the Power server. **Recommendation:** Avoid using an embedded Ethernet port for the SWT. Instead, use a non-embedded Ethernet NIC adapter for the SWT. Note that this is an IBM i iSCSI target recommendation only. It does not apply to the blade or System x iSCSI initiator.
Windows Server 2008 vs. Windows Server 2003	Both throughput and integrated server CPU utilization are essentially the same between these two operating systems. **Note**: VMware ESX was not included in the test configurations.

17.3 Test Configurations

The test results in this chapter were achieved using the following system configurations.

Test Config.	Description
IBMi71	This IBM i 7.1 configuration was used for tests involving a SWT or a SWI: Power 570, 9117-MMA (7380 processor feature, 4 cores, 4.7 GHz), rated at 21200 CPWs, IBM i 7.1, 16 GB memory, 1 5706 adapter (dual port -- used for 1 Gbps SWT), 2 573A adapters (used for 10 Gbps SWT), 2 573B adapters (used for 1 Gbps HWT), 4 572F SAS IOA (2 dual controller pairs), 60 SAS 433B disk drives, RAID 5 enabled, 4 parity sets (12 disks, 12 disks, 18 disks, 18 disks). One iSCSI attached server was a HS21 XM BladeCenter server with a copper iSCSI (p/n 26K6489) daughter card (for HWI testing) and built in Broadcom ports (for 1 Gb SWI testing). Another iSCSI attached server was a HS22 BladeCenter server with built in Broadcom BCM57710 NetXtreme II 10 GigE ports (for 10 Gb SWI testing). Performance tests were run using either Windows Server 2008 or Windows Server 2003 on the integrated servers. VMware ESX was not tested.
IBMi54	This IBM i 5.4 configuration was primarily used for HWI to HWT tests: System i Model 570 - 2-way 26F2 processor (7495 capacity card), rated at 6350 CPWs, IBM i 5.4, 40 parity protected (RAID 5) 4326 disks, 3 2780 disk controllers. The iSCSI attached server was a HS20 BladeCenter server with a copper iSCSI (p/n 26K6489) daughter card.

	Switches were Nortel L2/3 Ethernet (p/n 26K6524) and Cisco Intelligent Gigabit Switch. Performance tests were run using Windows Server 2003 on the integrated server. VMware ESX was not tested.

17.4 Effects of integrated server loads on the host system

Depending on the integrated server application activity, integrated server I/O operations impose an indirect load on the IBM i native CPU, memory and storage subsystems. The rest of this chapter describes some of the performance and memory resource impacts.

17.4.1 iSCSI Disk I/O Operations:

- The iSCSI disk operations use a scalable storage I/O access architecture. As a result, a single integrated server can scale to greater capacity by using multiple target and initiator iSCSI adapters to allow multiple data paths.
- In addition, there is no inherent partition cap to the iSCSI disk I/O. The entire performance capacity of installed disks and disk IOAs is available to iSCSI attached servers.
- The Windows disk drive "write cache" policy does not directly affect iSCSI operations. Write operations always "write through" to the host disk IOAs, which may or may not cache in battery backed memory (depending on the capabilities and configuration of the disk IOA).
- iSCSI attached servers use non-reserved IBM i virtual storage in order to perform disk input or output. Thus, disk operations use host memory as an intermediate read cache. Write operations are flushed to disk immediately, but the disk data remains in memory and can be read on subsequent operations to the same sectors.

  While the disk operations page through a memory pool, the paging activity is not visible in the "Non-DB" pages counters displayable via the WRKSYSSTS command. This doesn't mean the memory is not actively used, it's just difficult to visualize how much memory is active. WRKSYSSTS will show faults and paging activity if the memory pool becomes constrained, but some write operations also result in faulting activity.
  There are some integrated server operating system disk configuration rules you must take into account to enable efficient disk operations.

Operating System	Disk Configuration Rules
Windows	Windows disks should be configured as: • 1 disk partition per virtual drive. • File system formatted with cluster sizes of 4 kbyte or 4 kbyte multiples. • 2 gigabyte or larger storage spaces (for which Windows creates a default NTFS cluster size of 4kbytes). If necessary, you can use care to configure multiple disk partitions on a single virtual drive. • For storage spaces that are 1024 MB or less, make the partitions a multiple of 1 MB (1,048,576 bytes). • For storage spaces that are 511000 MB or less, the partition should be a multiple of 63 MB (66060288 bytes). • For storage spaces that are greater than 511000 MB, the partition should be a multiple of 252 MB (264,241,152 bytes).

VMware ESX	Generally speaking, the rules for aligning the storage depend on the operating system that runs in the virtual machine that uses the storage. • For example, if a storage space is assigned to a virtual machine that is running Windows, then the alignment rules defined above for Windows servers should be followed. • For a storage space that is assigned to a virtual machine that is running a non-Windows operating system, other alignment rules may apply. For more information, see the **Aligning storage partitions for VMware ESX Server on iSCSI attached integrated servers** document referenced at the end of this chapter.

These guidelines allow file system structures to align efficiently between the integrated server operating system and IBM i. They allow IBM i to efficiently manage the storage space memory, mitigate disk operation faulting activity, and thus improve overall iSCSI disk I/O performance.

Failure to follow these guidelines will cause iSCSI disk write operations to incur performance penalties, including page faults and increased serialization of disk operations.

- The CHGNWSSTG command and IBM Systems Director Navigator for i support the expansion of a storage space. After the expansion, the file system in the disk should also be expanded - but take care: don't create a new partition in the expanded disk free space (unless the new partitions meet the size guidelines above).

  Note: The Windows "DISKPART" command can be used to perform the file system expansion. However, it only actually expands the file system on "basic" disks. If a disk has been converted to a "dynamic" disk[1], the DISKPART command creates a new partition and configures a spanned set across the partitions. The second partition may experience degraded disk performance.

17.4.2 iSCSI virtual I/O shared data memory pool

Applications sharing the same memory pool with iSCSI disk operations may be adversely impacted if the iSCSI attached servers perform levels of disk I/O which can flush the memory pool. Thus, it is possible for other applications to begin to page fault because their memory has been flushed out to disk by the iSCSI operations. By default, the iSCSI virtual disk I/O operations occur through the *BASE memory pool.

In order to segregate iSCSI disk activity, iSCSI virtual disk I/O operations can be configured to run out of a shared data memory pool. The pool is enabled by creating a shared data memory pool and allocating at least 4096 kilobytes. The amount of memory required for the iSCSI memory pool depends on a number of factors, including the number of iSCSI attached servers and the expected sustained disk activity for all servers. For most environments, the recommended minimum iSCSI memory pool size is **512 MB**.

There are a couple of ways to create a shared data memory pool for use by all iSCSI attached servers:

- If you are using the **Create Server** Web GUI task to install a new integrated server, the GUI task provides an option to create a default iSCSI memory pool that is 512 MB.
- Otherwise, use the following command to create an iSCSI memory pool:
  CHGSHRPOOL POOL(*SHRPOOL*nn*) ACTLVL(*DATA) TEXT(*ISCSI) SIZE(524288)[2]
  where *nn* is the number of an unused shared data memory pool.
  You can use the WRKSHRPOOL command to see which memory pool numbers are available.

[1] Not to be confused with "dynamically linked" storage spaces.

[2] 524288 KB (512 MB) is the recommended minimum value, but 4096 KB is the absolute minimum supported.

Note that the *ISCSI text value is optional, but serves as a reminder that this memory pool is dedicated to iSCSI I/O. It also allows the **Create Server** Web GUI task to use this memory pool by default when creating new servers.

To use the iSCSI memory pool allocated above for an iSCSI attached integrated server:

- When installing a new iSCSI attached integrated server, select the memory pool allocated above in the Web GUI **Create Server** task or specify POOL(*SHRPOOL*nn*) on the **INSINTSVR** or **INSWNTSVR** command.
- If you already have an iSCSI attached server that uses the *BASE pool, select the memory pool allocated above in the Web GUI **Server Properties** task or specify POOL(*SHRPOOL*nn*) on the **CHGNWSD** command.

Notes:
- This iSCSI memory pool can be shared among all of the iSCSI attached servers on the system.
- If excessive page faults occur in the memory pool, increase the memory pool size to see if that helps.

17.5 IBM i memory rules of thumb for integrated servers

The IBM i machine pool memory "rule of thumb" is generally to size the machine pool with at least twice the active machine pool reserved size. Automatic performance adjustments may alter this according to the active load characteristics. But, there are base memory requirements needed to support the hardware and set of adapters used by the IBM i partition. You can refer to the IBM Systems Workload Estimator for estimates of these base requirements. The "rules of thumb" below estimates the additional memory required to support iSCSI.

The specific memory requirements of iSCSI attached servers vary based on many configuration choices, including the number of LUNs, number of iSCSI targets, number of NWSDs, etc.

- For most environments, use an iSCSI shared data memory pool with a minimum size of **512 MB**.
- In addition, a suggested minimum memory "rule of thumb"[3] for the machine pool and base pool is:

	For Each iSCSI Target	**For Each NWSD**
Machine Pool:	21 MB	1 MB
Base Pool:	1 MB	0.5 MB
Total (excluding shared pool):	22 MB	1.5 MB

Warning: To ensure expected performance and continuing machine operation, it is critical to allocate sufficient memory to support all of the devices that are varied on. Inadequate memory pools can cause unexpected machine operation.

[3] Based on a rough configuration of 5 LUNS per server, 2 VE connections per server, and two iSCSI target connections per server.

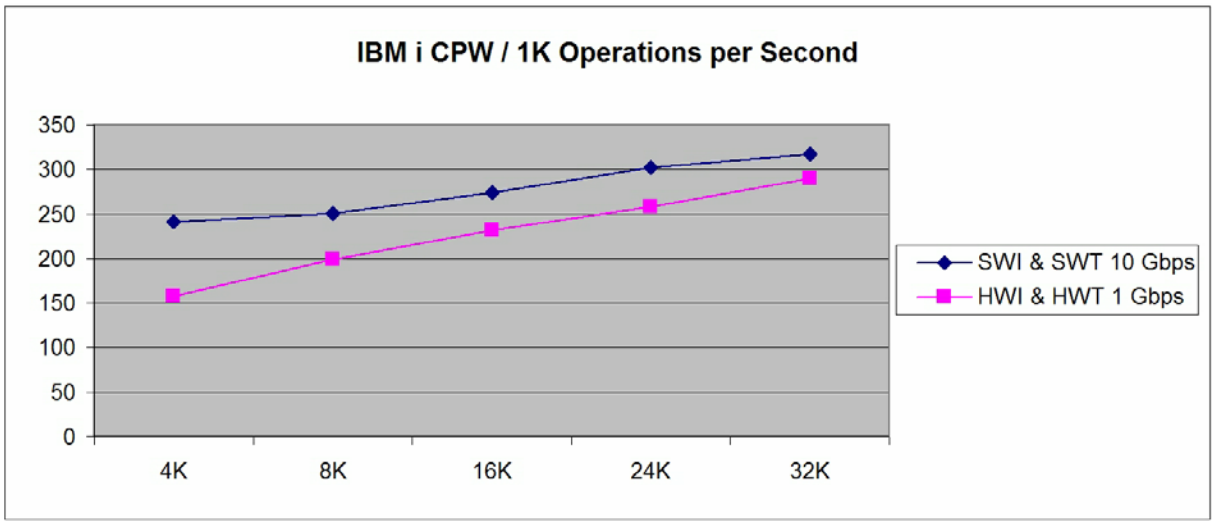
17.6 Disk I/O CPU Cost

Disk Operation Rules of Thumb	CPWs[4] / 1k ops/sec
iSCSI linked disks using a hardware target (HWT)	**230**
iSCSI linked disks using a software target (SWT)[5]	**275**

While the disk I/O activity driven by the iSCSI solution is not strictly a "CPW" type load, the CPW estimate is still a useful metric to estimate the amount of IBM i CPU required for a load. You can use the values above to estimate the CPW requirements if you know the expected I/O rate. For example, if you are using a software target and expect the integrated server to generate 800 disk ops/sec, you can estimate the CPW usage as:

275 CPWs/1kops * 800ops * 1kops/1000ops = 275 * 800/1000 = 220 CPWs

These rules of thumb are estimated from the results of performing file serving or application types of loads. In more detail, the chart below indicates an approximate amount of host processor (in CPW) required to perform a constant number of disk operations (1000) of various sizes. You can reasonably adjust this estimate linearly for your expected I/O level.



The chart shows the relative cost when performing operations with the following characteristics.

- 4K-32K block sizes
- A 33% random write, 67% random read mix of operations.

[4] A CPW is the "Relative System Performance Metric" from Appendix C. Note that the I/O CPU capacities may not scale exactly by rated system CPW, as the disk I/O doesn't represent a CPW type of load . This calculation is a convenient metric to size the load impacts. The measured CPW cost will actually decrease from the above values as the number of processors in the NWSD hosting partition increases, and may be higher than estimated when partial processors are used.

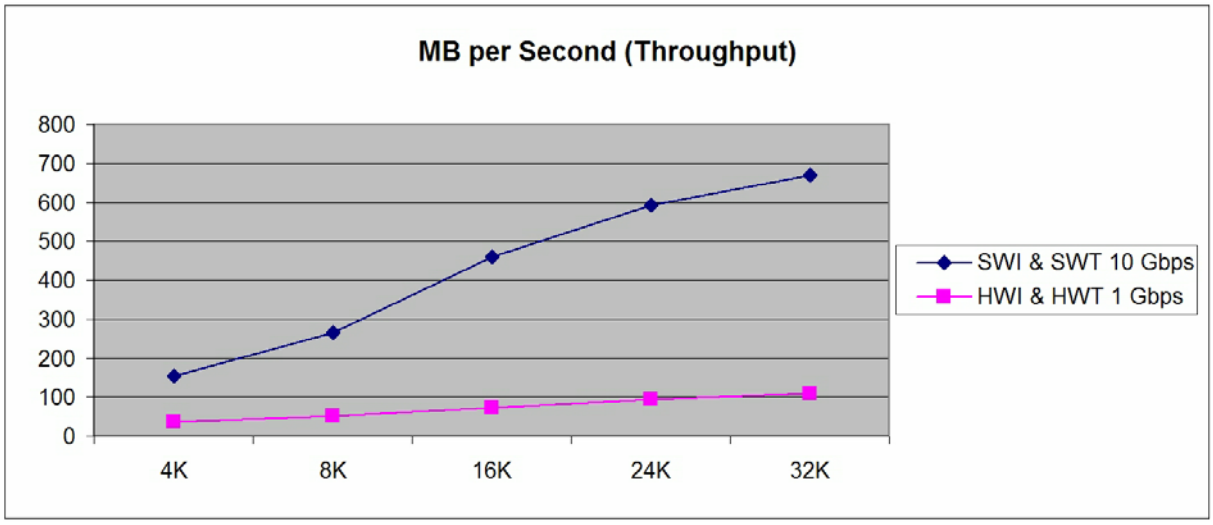
[5] The SWT CPW value is achieved using a 512 MB memory pool, disabled iSCSI digests, and using jumbo frames.
If these recommendations are not followed, then the SWT CPW cost could be much higher than the value shown.

- The SWI & SWT 10 Gbps configuration (IBMi71) had iSCSI digests disabled and used jumbo frames.
- The HWI & HWT 1 Gbps configuration (IBMi54) had iSCSI digests enabled and used standard frames.

On average, the SWI & SWT 10 Gbps CPW cost per 1K operations per second is about **20% higher** than the HWI & HWT 1 Gbps CPW cost.

17.7 Disk I/O Throughput

The chart below compares the throughput performance characteristics of the software based iSCSI solution (SWI & SWT 10 Gbps) against the hardware based iSCSI solution (HWI & HWT 1 Gbps). The charts indicates an approximate capacity of a single iSCSI target adapter when running various sizes and types of random operations.



The chart shows the throughput when performing operations with the following characteristics.

- 4K-32K block sizes
- A 33% random write, 67% random read mix of operations.
- The SWI & SWT 10 Gbps configuration (IBMi71) had iSCSI digests disabled and used jumbo frames.
- The HWI & HWT 1 Gbps configuration (IBMi54) had iSCSI digests enabled and used standard frames.

This chart indicates that the 10 Gbps software solution (SWI & SWT) can achieve **4-6 times** the throughput of the 1 Gbps hardware solution (HWI & HWT) for a given block size. As with all performance analysis, the actual values that you will achieve are dependent on a number of variables including workload, network traffic, etc.

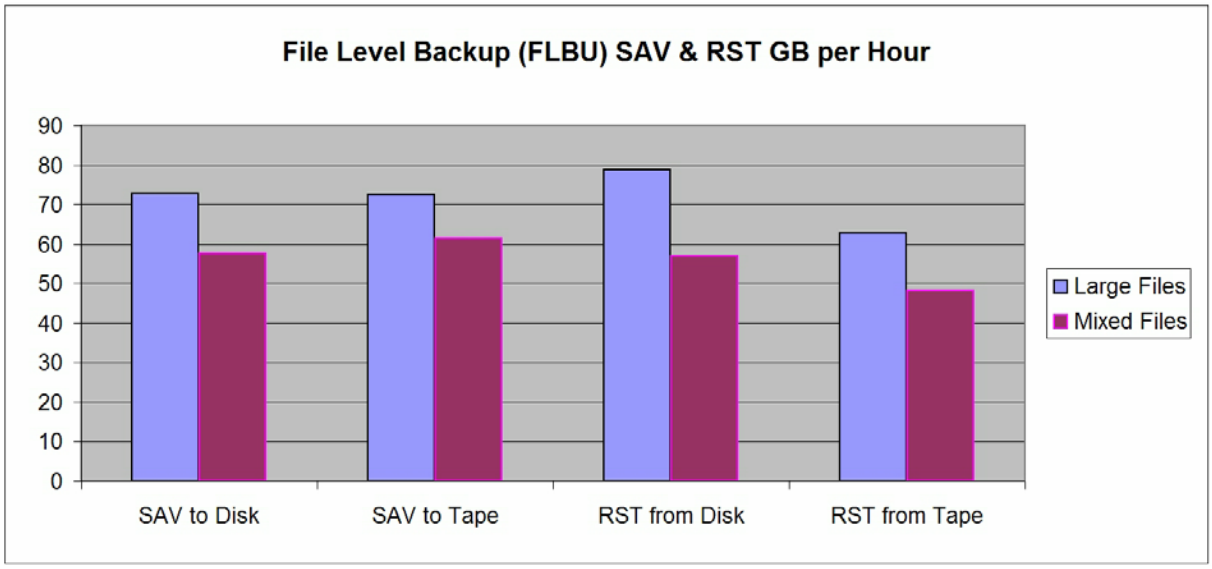
17.8 File Level Backup (FLBU) Performance for integrated Windows servers

The Integrated Server support allows you to save Windows server data (files, directories, shares, and the Windows registry) to tape, optical or disk (*SAVF) in conjunction with your other IBM i data. That is, this “file level backup” approach saves or restores the Windows files on an individual basis within the stream of other IBM i data objects. It’s not recommended that this approach is used as a primary backup procedure. Rather, you should still periodically save your storage spaces and the NWSD associated with your Windows server for disaster recovery.

Saving individual files does not operate as fast as saving the entire storage spaces. The save of a storage space on a equivalent machine and tape is about 210 Gbytes per hour, compared to the approximately 70 Gbytes per hour achieved below.

The chart below compares some SAV and RST rates for iSCSI attached servers. These results were measured using the IBMi54 test configuration. The target tape drive was a model 5755-001 (Ultrium LTO 2). All tests were run with jumbo frames enabled.

The legend label “Mixed Files” indicates a save of many files of mixed sizes - equivalent to the save of the Windows system disk. “Large files” indicates a save of many large files - in this case many 100MB files.



17.9 Summary

The IBM i iSCSI solution provides scalable integration for full file, print and application servers running selected Windows Server 2008, Windows Server 2003, or VMware ESX editions. They provide flexible consolidation of IBM i solutions and integrated server services, in combination with improved hardware control, availability, and reduced maintenance costs. These solutions perform well as a file or application server for popular applications, using the IBM i host disks. The iSCSI solution provides integrated server configuration flexibility and performance scalability. As part of the preparation for integrated server installations, care should be taken to estimate the expected workload of the integrated server applications and reserve sufficient IBM i resources for the integrated servers.

17.10 Additional Sources of Information

Web site: "IBM i integration with BladeCenter and System x" at
http://www.ibm.com/systems/i/advantages/integratedserver/

Online documentation: "IBM i integration with BladeCenter and System x" at
http://publib.boulder.ibm.com/iseries/
Choose V7R1. In the "Contents" panel expand "IBM i 7.1 Information Center". Then expand "Blade and System x".

PDF: "iSCSI Initiator Hardware Configuration" at
http://www.ibm.com/systems/resources/initiator_hw_config.pdf

PDF: "Aligning storage partitions for VMware® ESX Server on iSCSI attached integrated servers" at
http://www.ibm.com/systems/resources/systems_i_advantages_integratedserver_pdf_vmware_storage_alignment.pdf

Redbook: "Implementing Integrated Windows Server through iSCSI to System i5", SG24-7230 at
http://www.redbooks.ibm.com/abstracts/sg247230.html

Redbook: "Tuning IBM System x Servers for Performance", SG24-5287 at
http://www.redbooks.ibm.com/abstracts/sg245287.html
While this document doesn't address integrated server configurations specifically, it is an excellent resource for understanding and addressing performance issues with integrated servers.

Chapter 18. Blade Performance

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

18.1 VIOS and JS12 Express and JS22 Express Considerations

Most of our work consisted of measurements with the JS22 offering and external disks using the DS4800 product. The following are results obtained in various measurements and then a few general comments about configuration will follow.

18.2 BladeCenter H JS22 Express running IBM i operating system/VIOS

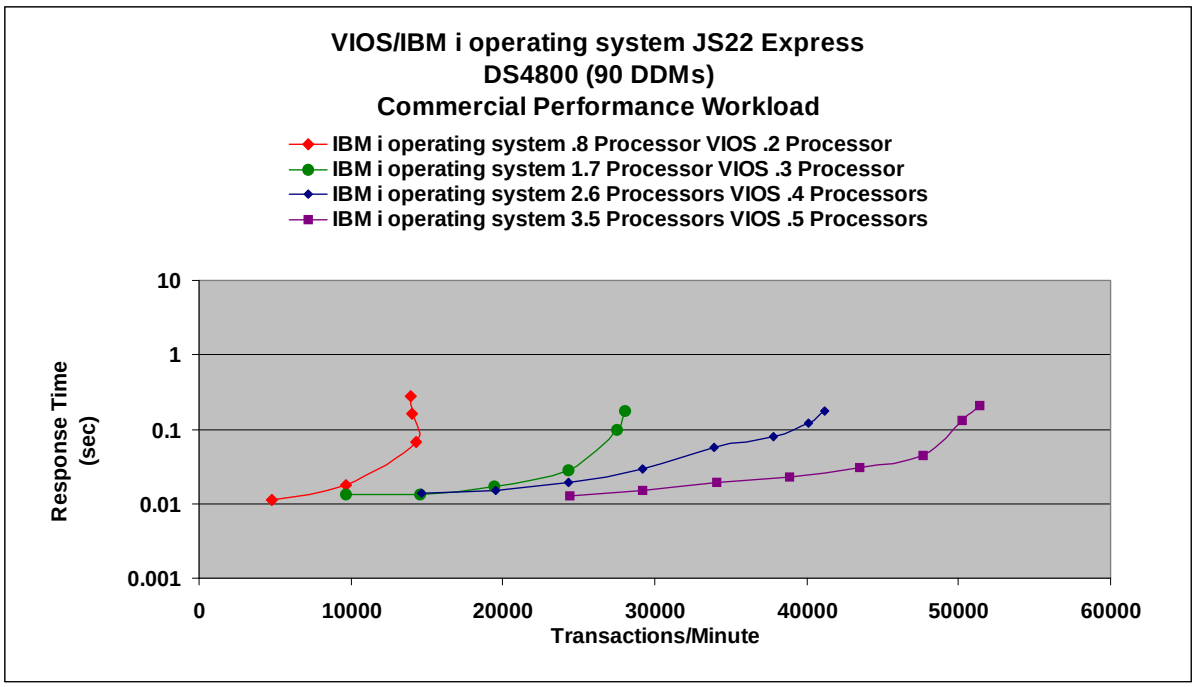
The following tests were run using a 4 processor JS22 Express in a BladeCenter H chassis, 32 GB of memory and a DS4800 with a total of 90 DDMs, (8 DDMs using RAID1 externalized in 2 LUNs for the system ASP, 6 DDMs in each of 12 RAID1 LUNs (a total of 72 DDMs) in the database ASP, and 10 DDMs unprotected externalized in 2 LUNs for the journal ASP). We had two Fibre Channel attachments to the DS4800 with half of the LUNs in each of the ASP's using controller A as the preferred path and the other half of the LUNs using controller B as the preferred path. The following charts show some of the performance characteristics we observed running our Commercial Performance Workload in our test environment. Your results may vary based on the characteristics of your workload. A description of the Commercial Performance Workload can be found in appendix A of the Performance Capabilities Reference.

Creating and running multiple LPARs can lead to unique system management challenges. Reference. The following is a link to an LPAR white paper.
http://www.ibm.com/systems/i/solutions/perfmgmt/pdf/lparperf.pdf .

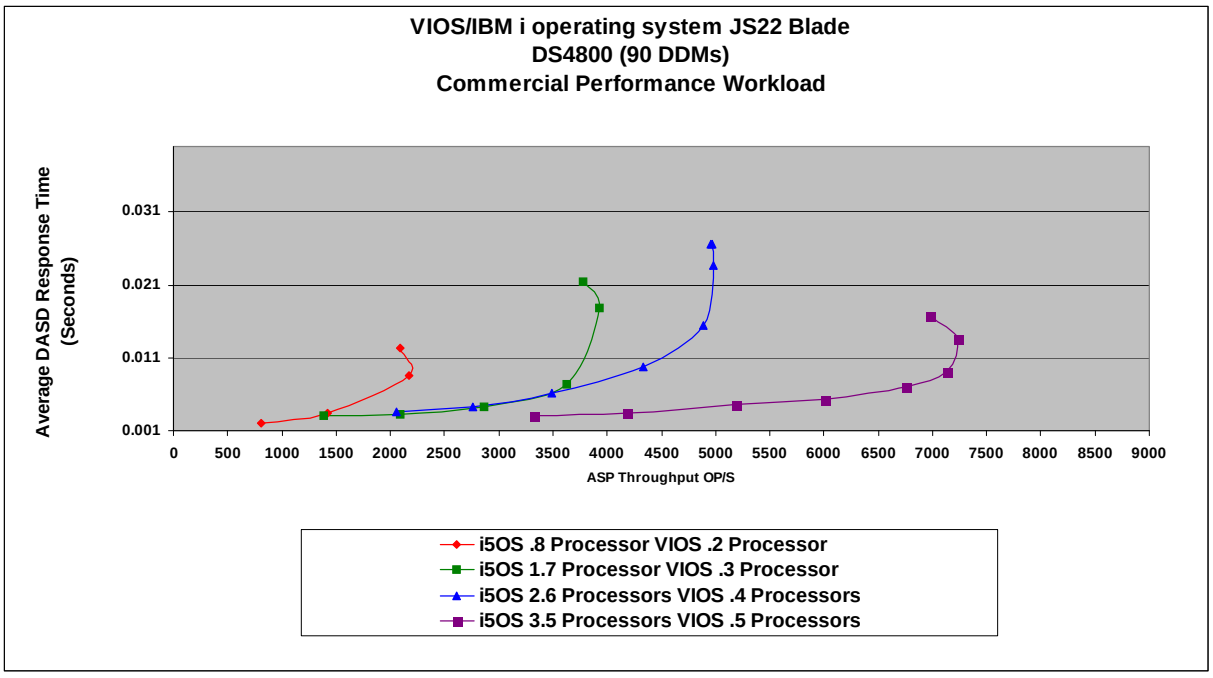
For most of our testing we only utilized one IBM i operating system partition on our JS22 Express. Note that VIOS is the base operating system on the JS22 Express, installed on the internal SAS Disk and VIOS must virtualize the DS4800 LUNs and communication resources to the IBM i operating system partition, which resides on the DS4800 DDMs.

VIOS/IVM must have some of the memory and processor resources for this vrtulization. The amount of resources needed will be dependent on the physical hardware in the Blade Center and the number of partitions being supported on a particular Blade. For our testing we found that we could not operate with under 1 GB of memory and for all of the tests in this section we used 2 GB of memory. The number of processors varied for each experiment and the charts will define the processors used in that experiment.

One important thing to note is that we only changed the amount of memory and processors in the VIOS partition. Otherwise the rest of the settings for the VIOS partition are as they default when the basic configurations is created during the VIOS install. So the VIOS partition processors in my experiments are always set up as shared, only the IBM i operating system partition is created using dedicated processors.



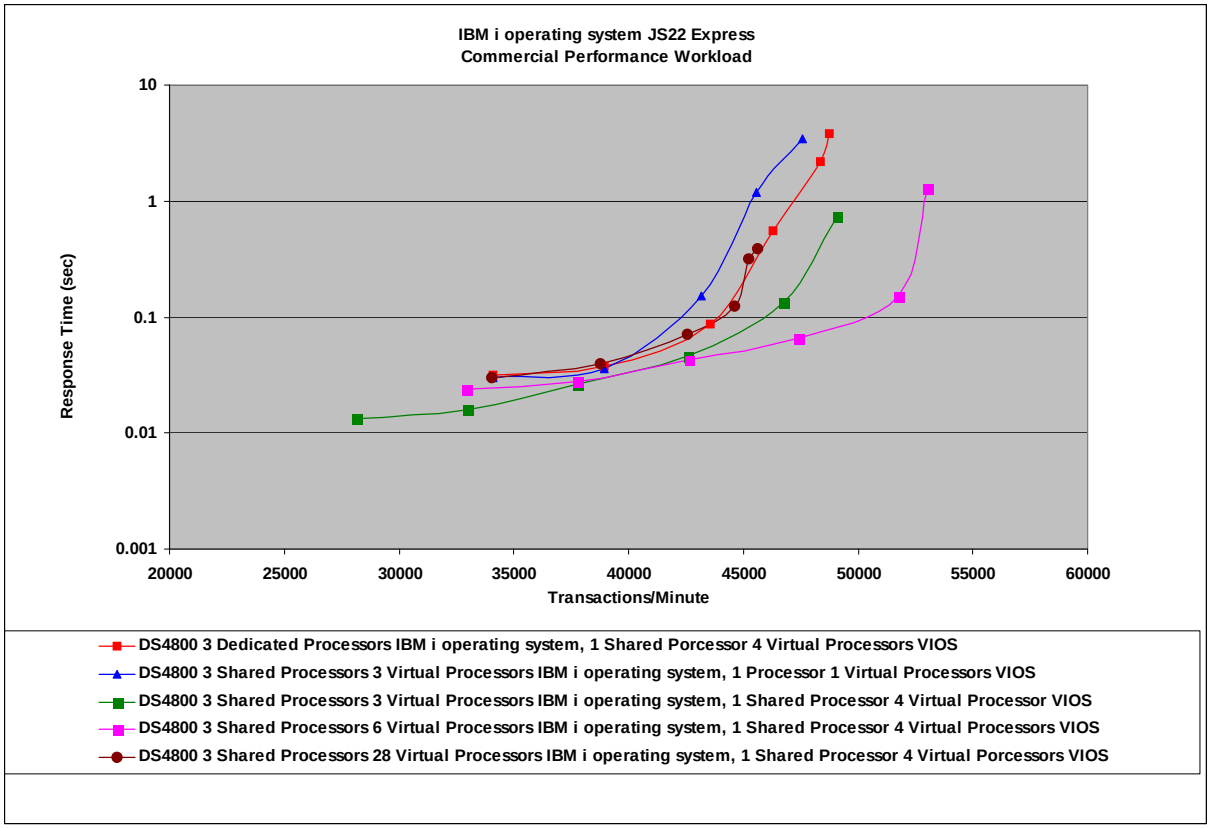
The chart above shows some basic performance scaling for 1, 2, 3 and 4 processors. For this comparison both partition measurements were done with the processors set up as shared, and with the IBM i operating system partition set to capped. The rest of the resources stay constant, which consists of 90 RAID1 DDMs in a DS4800 under 16 LUNs 2 GB of memory assigned to VIOS and 28 GB assigned to the IBM i operating system partition. Note that only 1 LPAR is running at the time of the experiment.



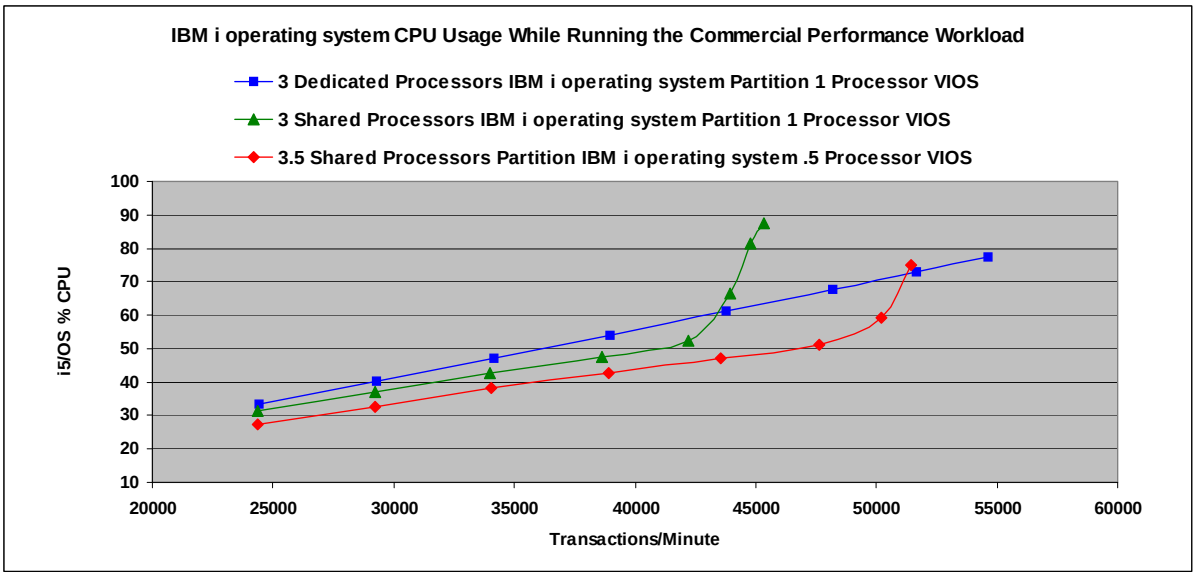
The following charts are a view of the characteristics we observed during our Commercial Performance Workload testing on our JS22 Express. The first chart shows the effect on the Commercial Performance Workload when we apply 3 Dedicated processors and then switch to 3 shared processors. Then incremented the number of virtual processors available.

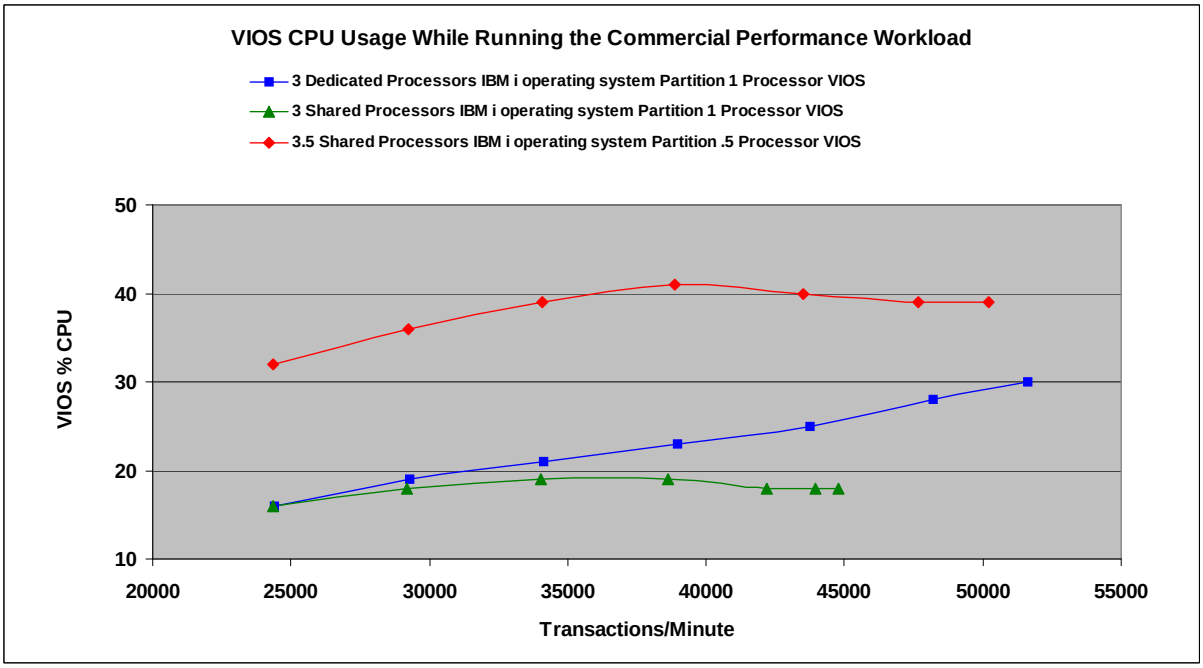
The “red line” is our dedicated processor set up, which is our baseline. The “blue line” is turning on shared processors in what we might have thought of as a fair comparison where 1 virtual processor was assigned for each real processor, resulting in 1 virtual processor for VIOS and 3 virtual processors for IBM i operating system. The Next experiment the “green line” was to increase the number of virtual processors assigned to VIOS but not the number of virtual processors assigned to IBM i operating system. Four virtual processors assigned to VIOS seemed to worked best for our environment. Next was to increase the number of virtual processors assigned to the IBM i operating system environment. Six virtual processors seen in the “purple line” optimized our environment best. As we increased from 6 virtual processors we started losing performance until we had increased to the 28 virtual processors available to me shown in the “dark red line”

Not all workloads will react in the same way but it is important to note that a small change to your configuration can have a large influence on your performance positive and negative. .

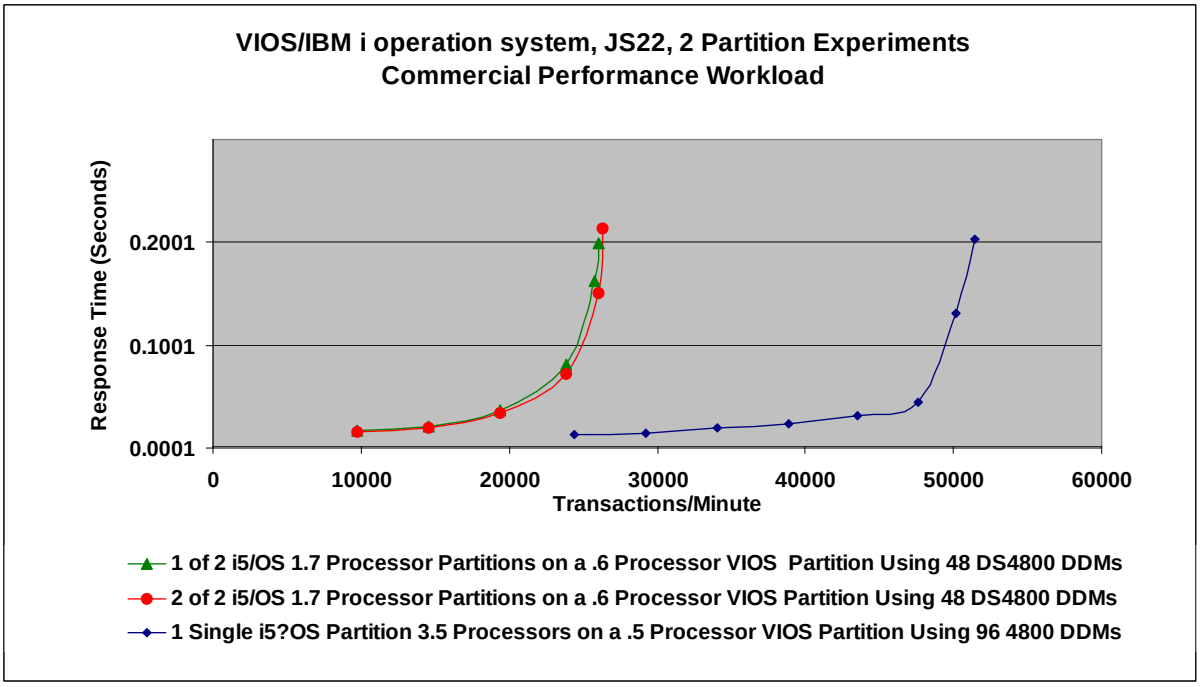


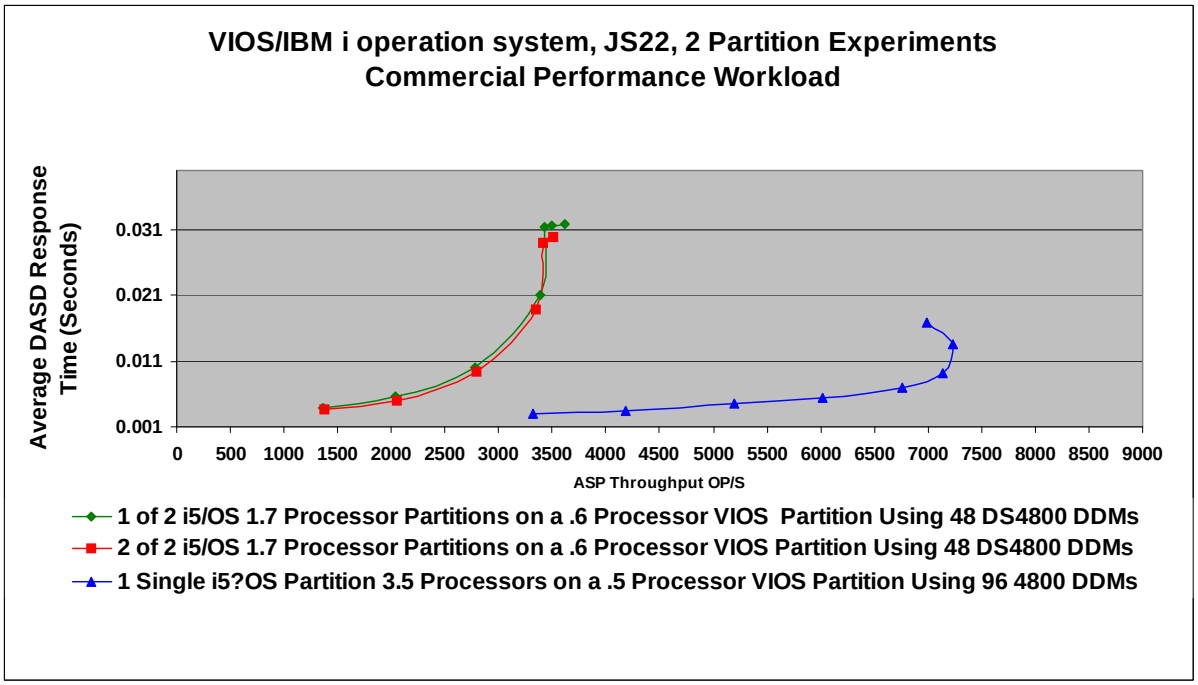
In following single partition Commercial Performance Workload runs the average VIOS CPU stayed under 40%. So we seem to have VIOS resource available but in a lot of customer environments communications and other resources are also running and these resources will also be routed through VIOS.





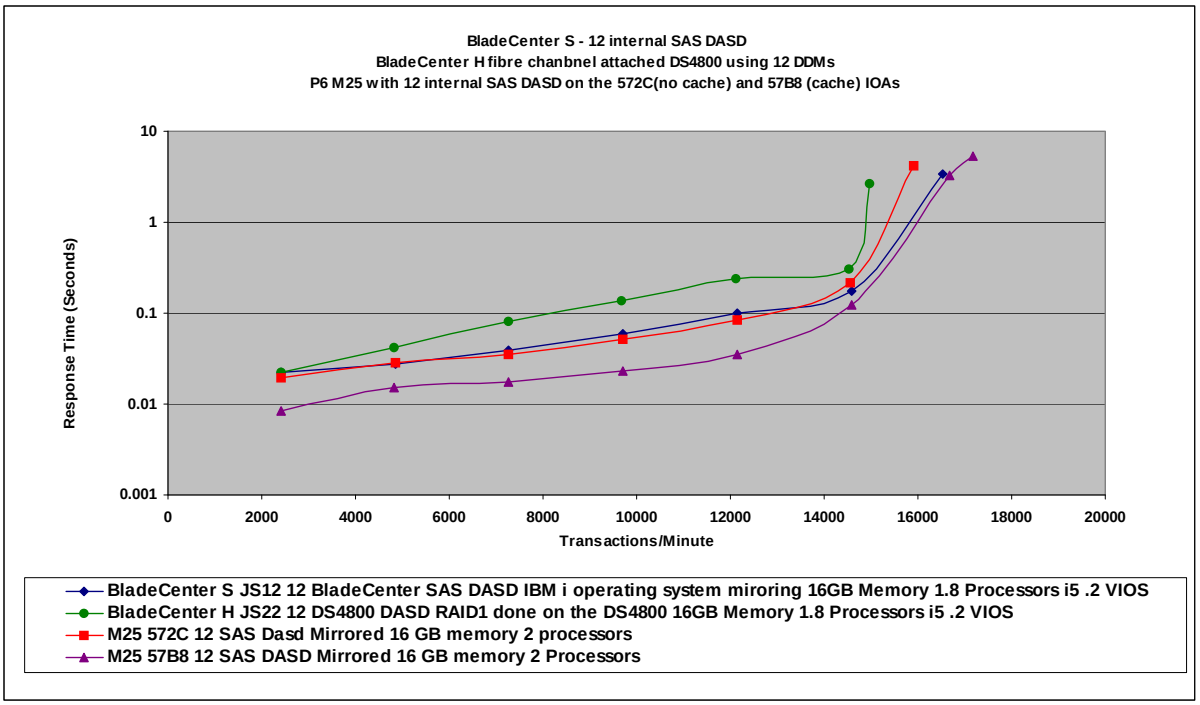
The following chart shows two IBM i operating system partitions using 14GB of memory and 1.7 processors each served by 1 VIOS partition using 2GB of memory and .6 processors. The Commercial Performance Workload was running the same amount of transactions on each of the partitions for the same time intervals. Although there is an observed cost for VIOS to manage multiple partitions, VIOS was able to balance services to the two partitions. Experimenting with the number of processors and memory assigned to the partitions might yield a better environment for other workloads.

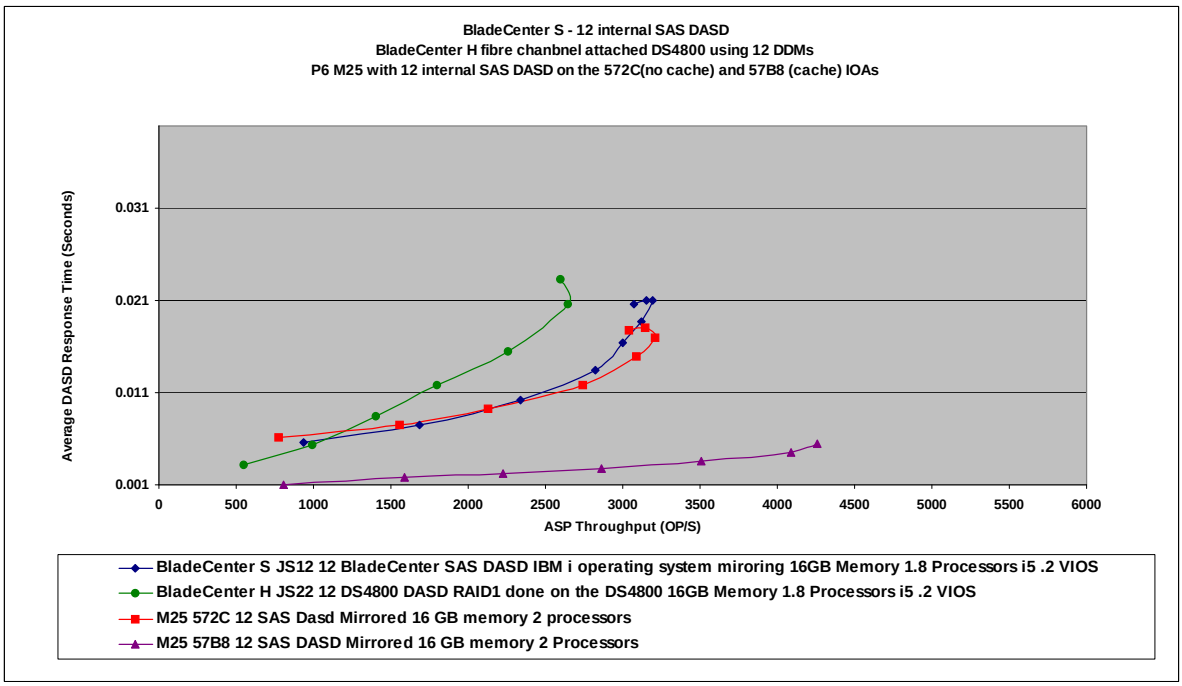




18.3 BladeCenter S and JS12 Express

The IBM i operating system is now supported on a JS12 Express in a BladeCenter S. The system is limited to 12 SAS DASD and the following charts try to characterize the performance we achieved during experiments with the Commercial Performance Workload in the IBM lab. Using a JS22 Express in a BladeCenter H connected to a DS4800, we limited the resources in order to get a comparison to the SAS DASD used in the BladeCenter S.





18.4 JS12 Express and JS22 Express Configuration Considerations

1. On blades, using the IVM interface, customers are limited to a single VSCSI and 16 logical units (which IBM i operating system perceives as if they were physical drives). Many customers will want to deploy between 12 and 16 LUNs and maximize symmetry. Consult carefully with your support team on the choices here. This is the most important consideration as it is difficult to change later. Consult also any available Best Practices manuals for a given SAN attached storage server.

NOTE: It is possible for blade-based customers to use the VIOS command line interface to create more VSCSI interfaces and map 16 LUNs to each VSCSI created. See VIOS configuration manuals for information on creating VSCSI and mapping LUNs from the command line.

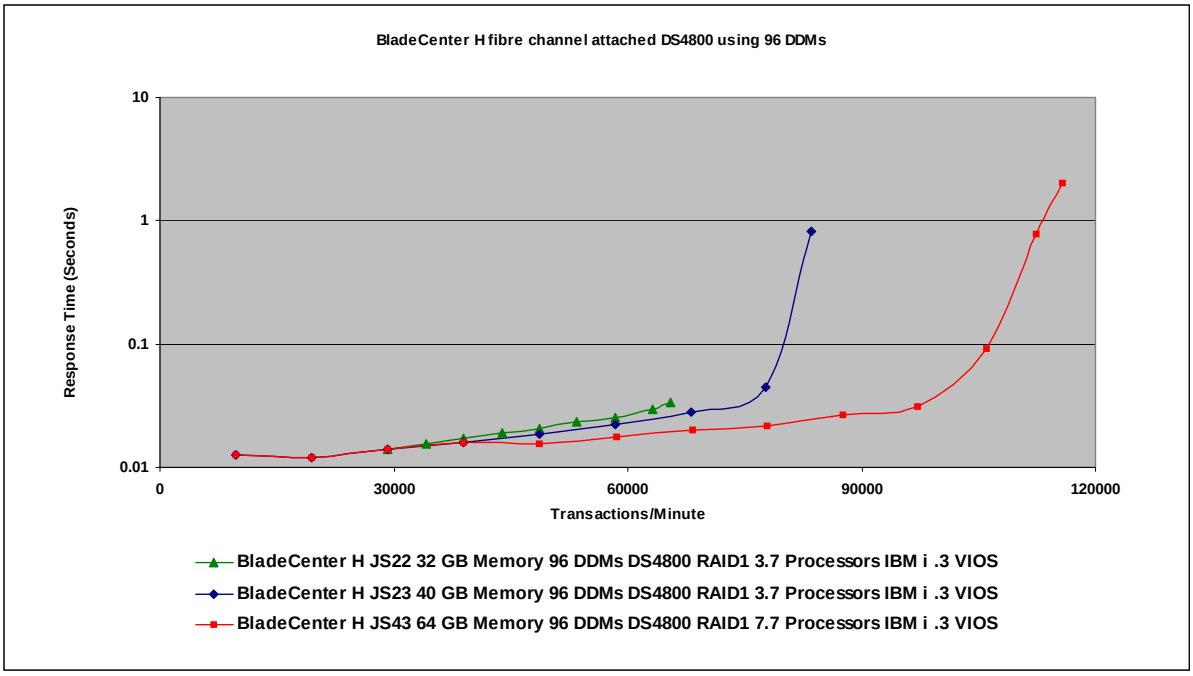
2. The VIOS partition should be provided with between 1 and 2 GB of memory for disk-based usage's. If virtual LAN is a substantial factor, more memory may be required.

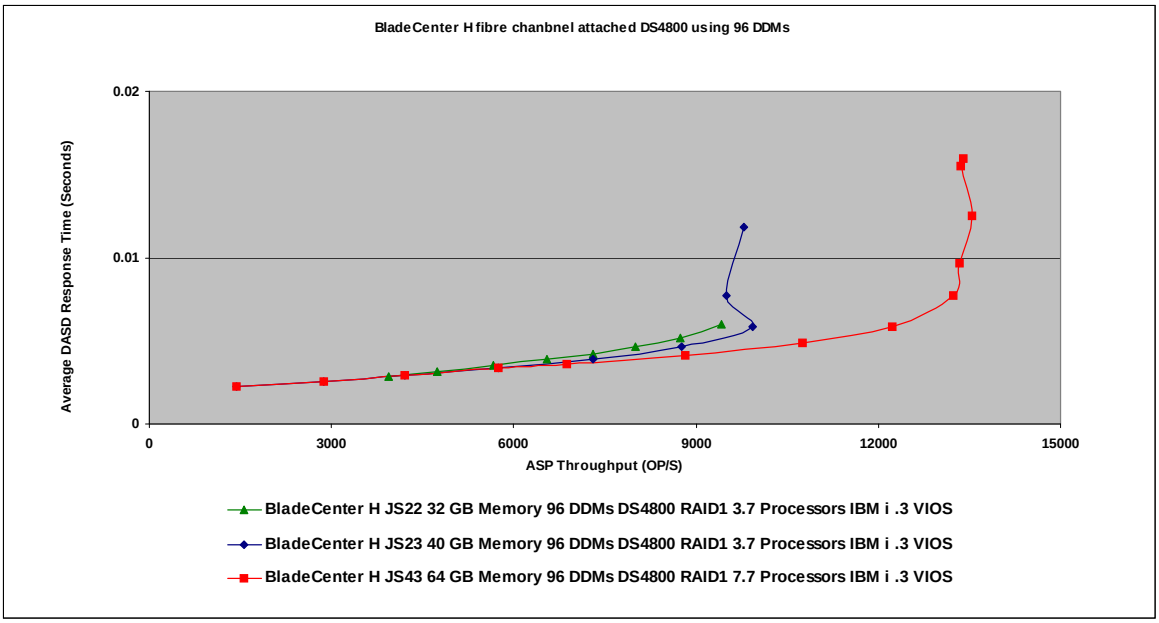
18.5 DS3000/DS4000 Storage Subsystem Performance Tips

Physical disks can be configured various ways with RAID levels, number of disks in each array and number of LUNs created over those arrays. There are also various reasons for the configurations that are chosen. One end user might be looking for ease of use and choose to create one array with multiple LUNs, where another end user might consider performance to be a more critical issue and select to create multiple arrays. The following charts are meant to show possible performance affects of various configurations using the Commercial Performance Workload.

18.6 BladeCenter S and BladeCenter JS23 and JS43

Below are some comparison charts of the JS23 and JS43 along with the JS22. The JS22 was only allowed 32 GB of memory and the workload could not drive the CPU any further with the available memory. The JS23 and JS43 have more memory available. The charts list the amount of memory needed to achieve the workload results we observed.



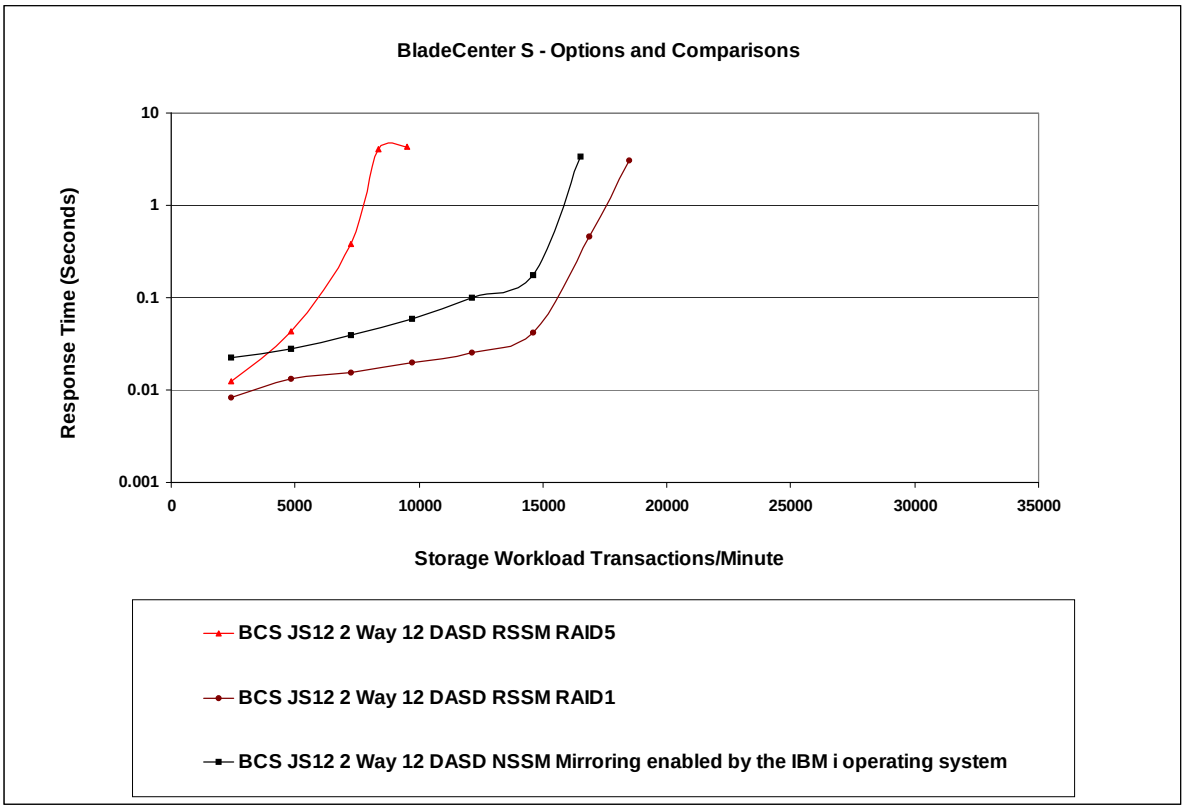


18.7 BladeCenter S RAID SAS Switch Module

The following section will try to show some side by side comparisons with a few of the available options and a couple comparisons to the current P6 550 model with the 572A IOA for reference. These results are based on the Commercial Storage Workload and are not an attempt to analyze cost competitiveness only observed performance differences of the options we chose to explore in these experiments.

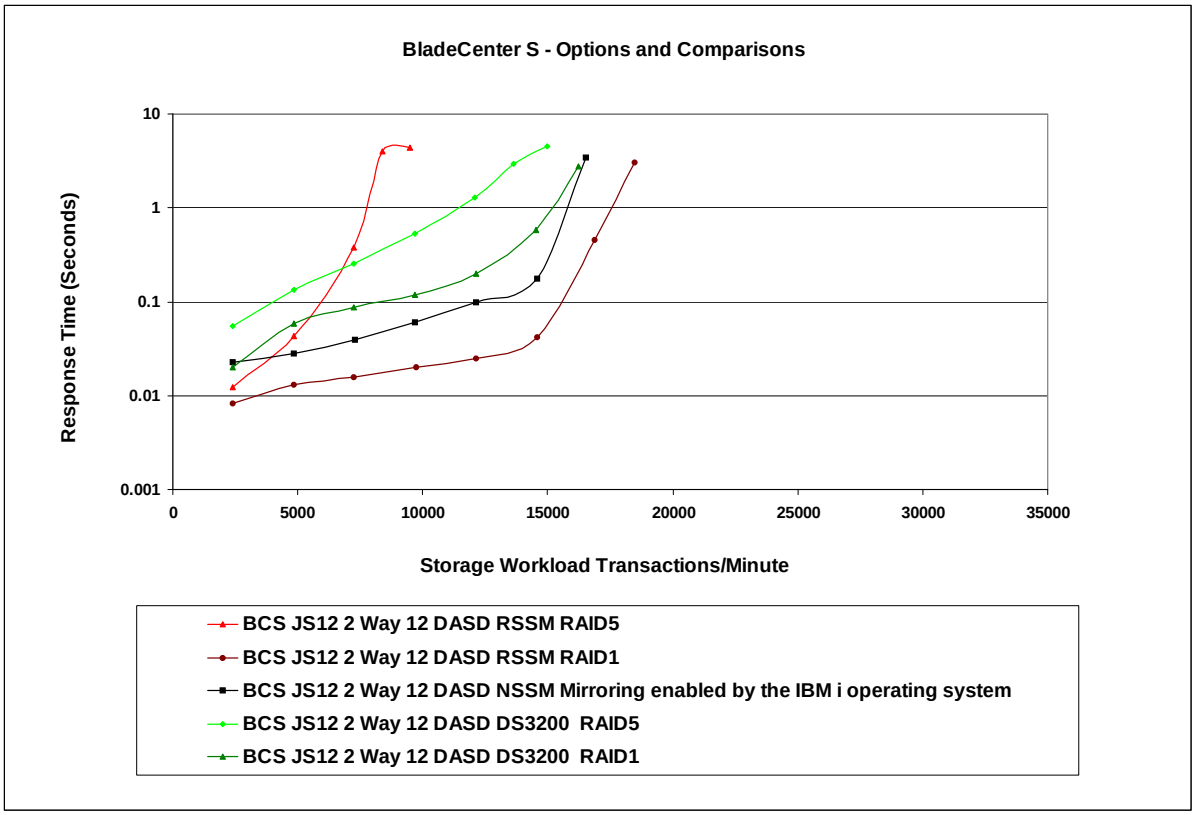
As we reference the charts we will be focusing on the workload transactions per minute charts. You can find the ASP op/s chart that corresponds to the workload transaction charts at the end of this section.

The first chart shows the BladeCenter S with the Non-Raid SAS Switch Module (NSSM) and the new Raid SAS Switch Module (RSSM).



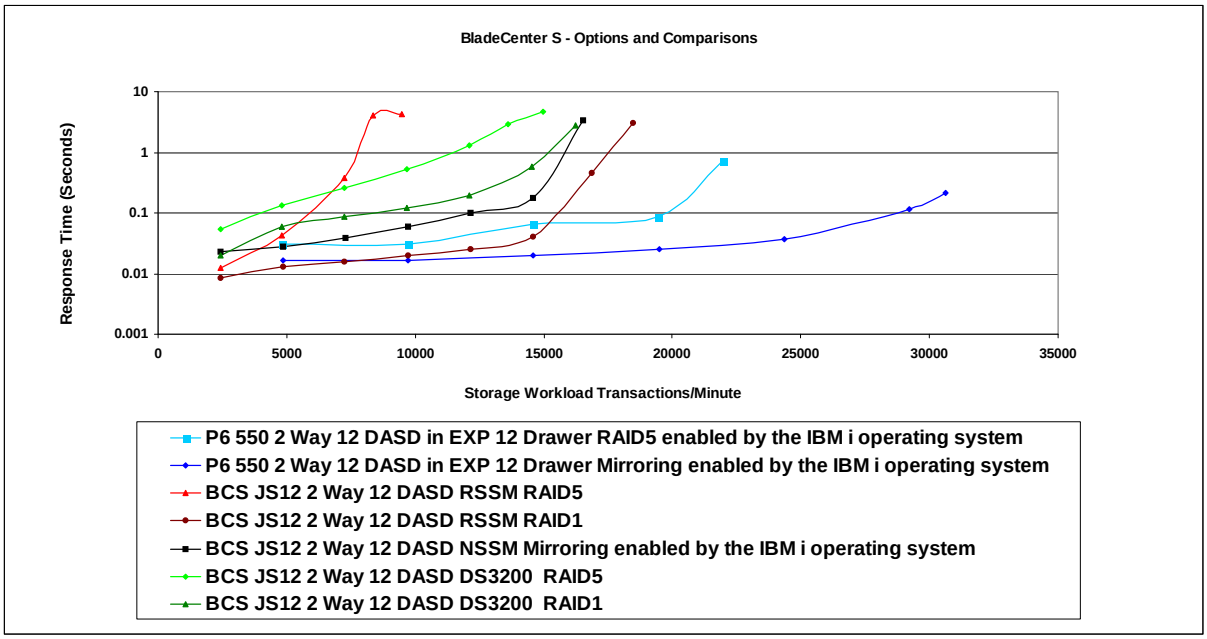
The new RSSM's enable RAID5 as well as RAID1 configurations. The benefits of the new hardware can be seen when comparing the current NSSM with IBM i controlling the mirroring and the RSSM set up as RAID1. As can also be from this chart RAID5 has a performance cost which some workloads can tolerate, but also keep in mind that RAID5 offers more capacity.

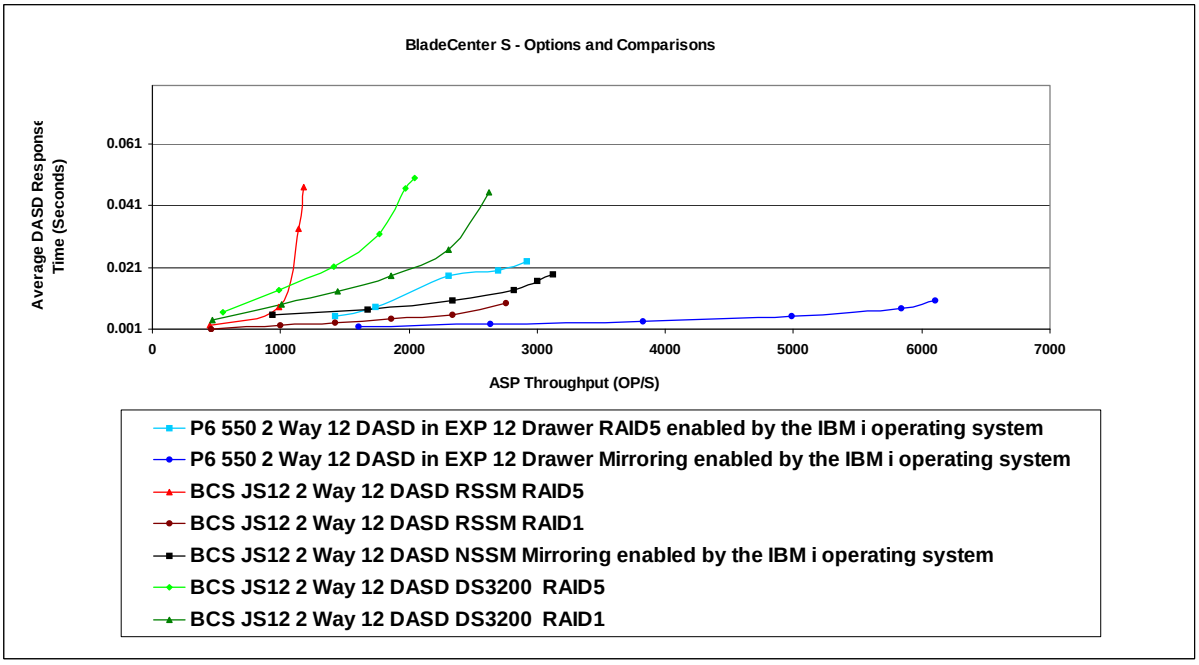
Another environment we experimented with was external storage such as the DS3200 on the BCS compared to the BCS internal DASD options. Below you will find the previous chart with the addition of the DS3200.



The RAID5 option on the DS3200 offers a little more performance than the RSSM RAID5 option but again this is a performance view not a cost analysis. As seen here the RAID1 and IBM i mirroring options still offer the best performance. The next aspect to introducing the DS3200 into this environment is expandability. Currently only 6 DASD units are offered internally on the RSSM or the NSSM. The DS3200 can be expanded to 48 DDMs. So if short term growth will be an issue then the DS3200 does offer more flexibility there. If RAID5 is a necessary element for the workload environment then more DDMs in a DS3200 configured to RAID5 may get to an expectable RAID5 performance level. Again this is not a cost comparison but with the performance information here it may help in doing that cost comparison.

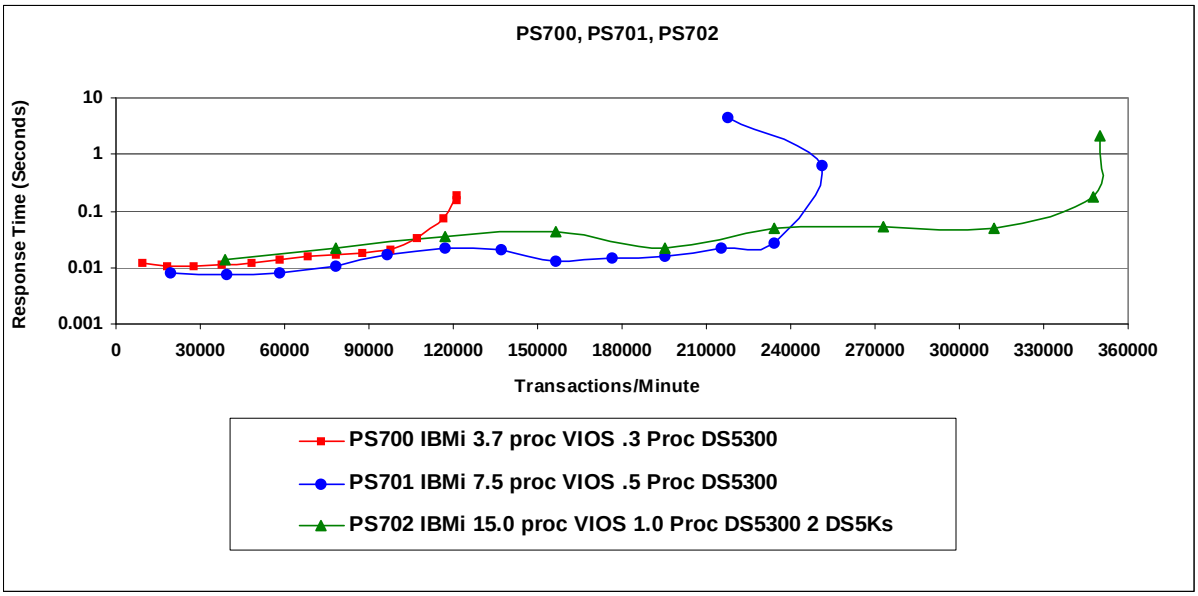
The final chart in this section places a P6 550 into the mix. How does the performance of the BladeCenter and the storage options compare to current IBM i Power 6 stand alone systems. The P6 550 has the option of having the 57B8/57B7 IOA built in and can have 6 DASD in the unit itself and can attach an EXP12 drawer to expand the DASD on that IOA to 18. The Storage workload was run on a system with 12 DASD in an EXP12 drawer attached and no DASD in the unit.

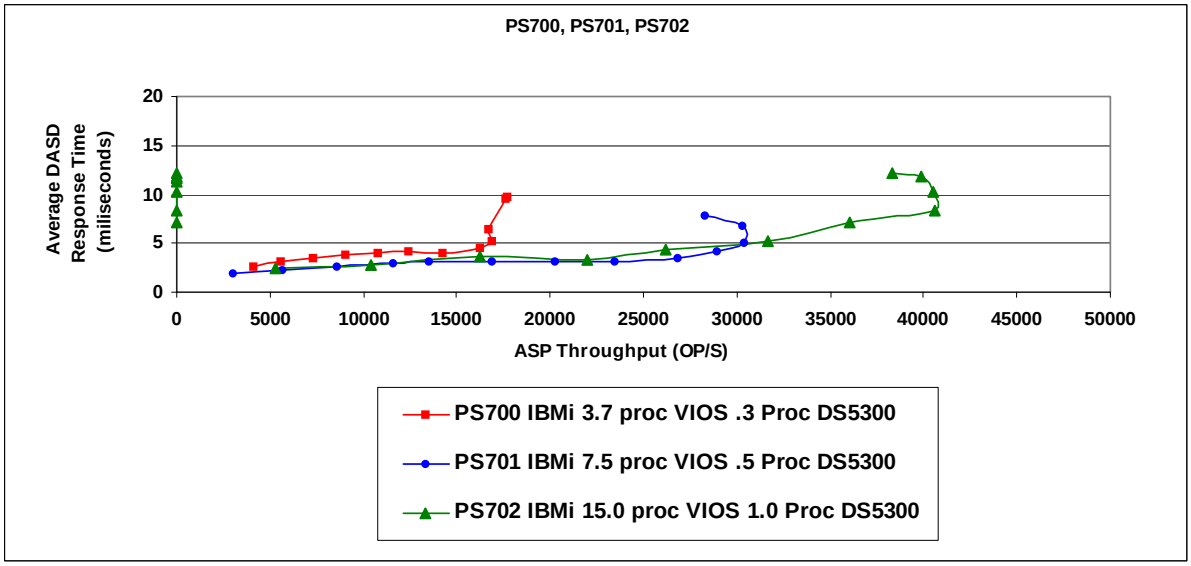




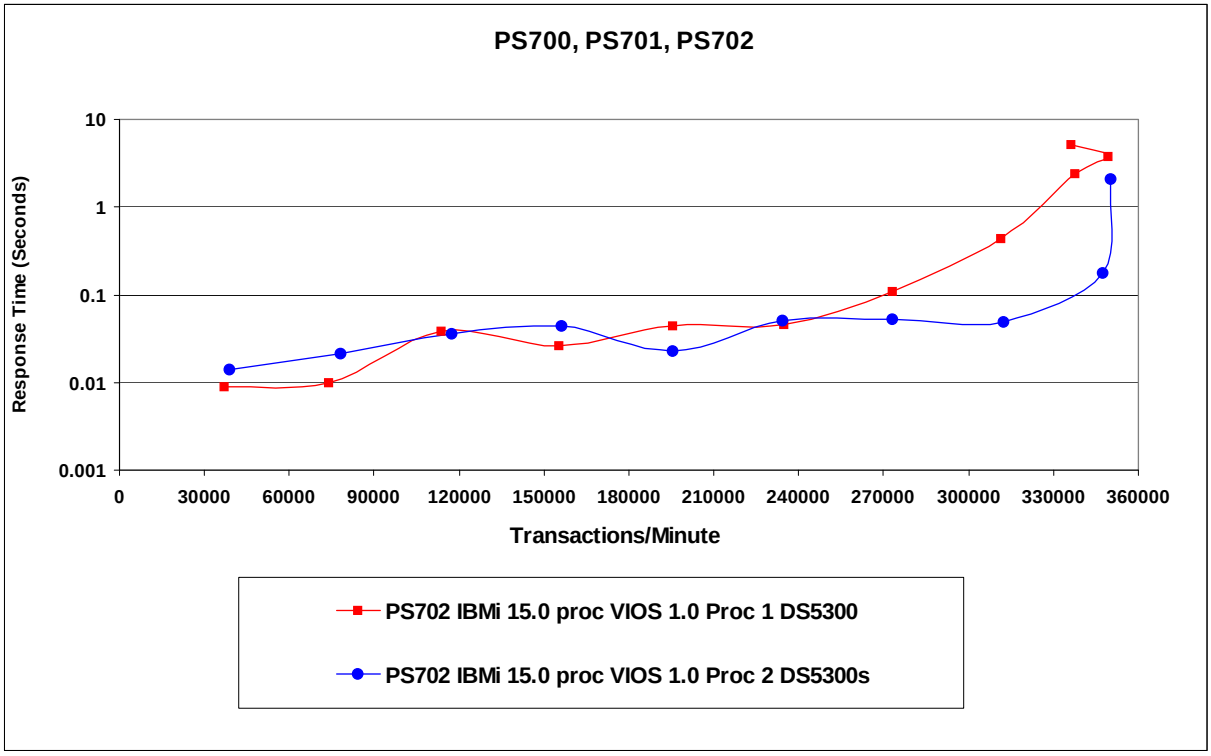
18.8 PS700, PS701, and PS702

The new blade models offer performance improvements over previous models, also offering a 16 processor system. The following charts were created from data collected during IBM Rochester lab experiments using the Commercial Performance Workload. The CPW rating for the different blade offerings can be found in the appropriate chapter in this document. The information here is extrapolated from collection services information at various points in the workload run. DS5300 fiber channel attached storage units were used in the experiments. In order to get the desired response time to push the processors we added a second DS5300 to the PS702 measurements. The charts depicting that are found later in this section.

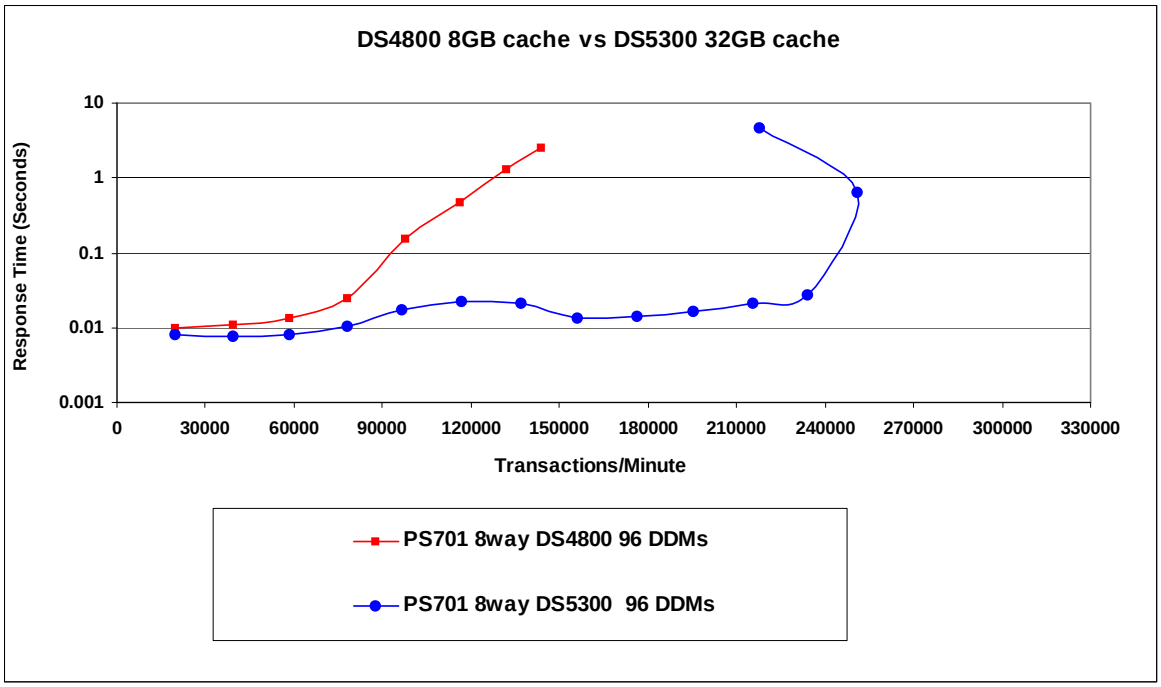


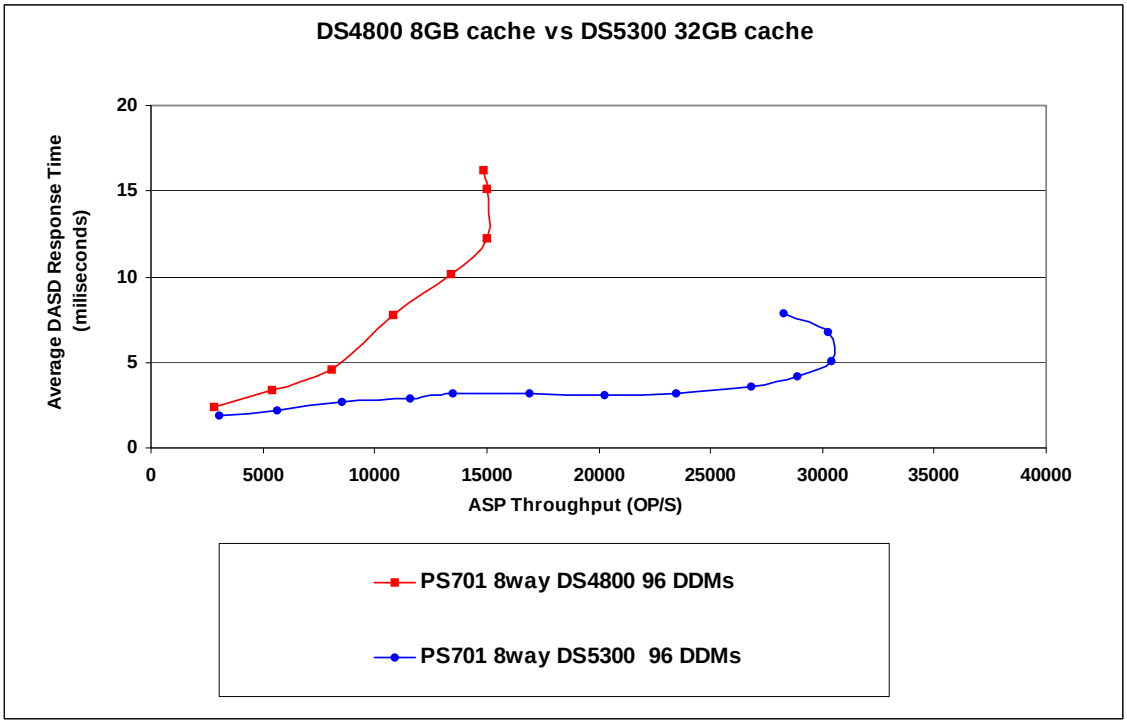


In the previous charts we added a second DS5300 to the BladeCenter for the PS702 measurements. In order to get a CPW rating for the system we attempt to drive the processors and to do this we try to make sure memory and DASD do not become the bottle neck. The following chart depicts the results of a run with a single DS5300 and a run with two DS5300 models. Each DS5300 model has 32 GB of cache and 256 physical 15K RPM DASD in 16 RAID 10 arrays. we created 32 LUNs on each DS5300 and assigned each LUN two paths in VIOS. The LUNs surface as Multipath devices to the IBM i partition. The queue depth at VIOS was set to 64 for each hdisk. As can be seen in the chart the experiment with one DS5300 was nearing the limit of the PS702's processors when we started getting the undesirable workload response times. Adding the second DS5300 did not get us beyond the peak that we saw from the processors but it did give us the desired response times so that we could see the bend in the workload and not the bend in the DS5300.

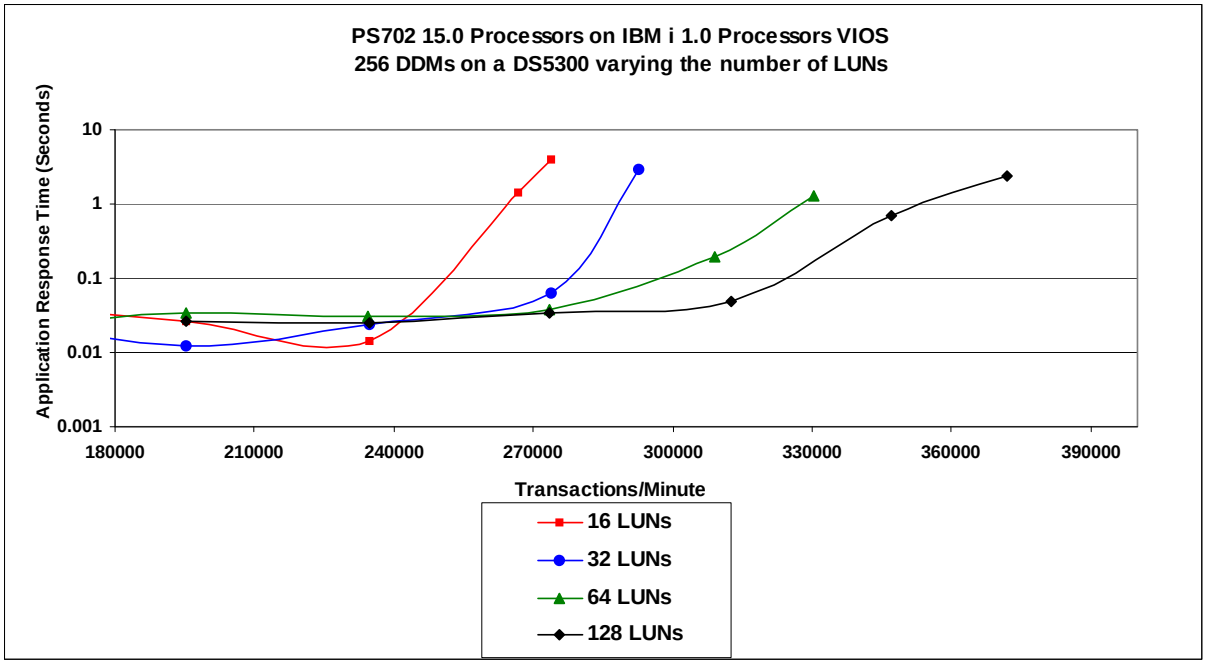


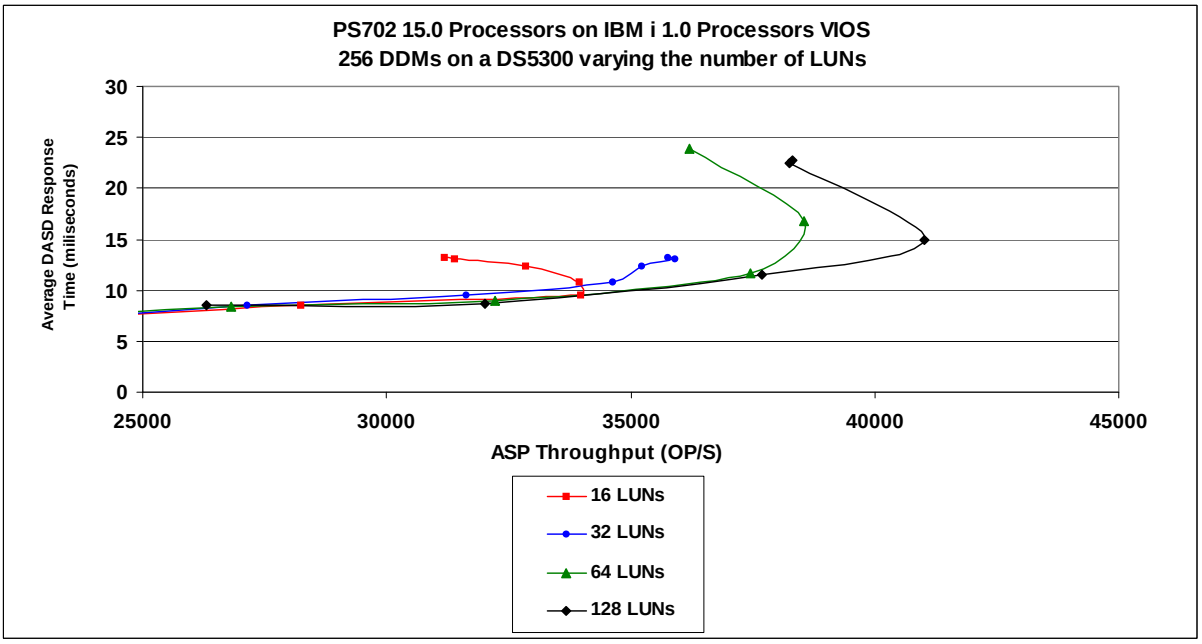
On previous blade experiments we used DS3200, DS3400 and DS4800 storage units and we wanted to make a comparison with the DS5300 and the DS4800 to show the performance difference available in the DS5300. The physical DDMs are both 15 RPM devices the DS4800 only had 4 GB of cache in each controller. The DS5300 has 16 GB of cache on each controller, In both configurations we set up 96 DDMs in RAID 10 arrays and surfaced them as 16 LUNs. As can be seen in the charts, the new DS5300 hardware with the larger cache offers a great performance improvement over the DS4800 for our DASD-intense workloads.





One of the questions that presents itself quite often is how does the number of LUNs presented to an IBM i partition affect performance. In the IBM Rochester lab we did some experiments attaching a DS5300 to a BladeCenter and varying the number of LUNs from the same 256 physical DDMs in the DS5300. The size of the LUNs was changed each time to make use of all storage available on the DS5300. The charts show a steady improvement to 128 LUNs after that point we found no substantial improvement.





Chapter 19. General Performance Information, Tips, and Techniques

Note: This chapter does not contain updated performance information beyond what was in the April/October 2011 version.

This section's intent is to cover a variety of useful topics that "don't fit" in the document as a whole, but provide useful things that customers might do or deal with special problems customers might run into on your IBM i system. It may also contain some general guidelines.

19.1 Adjusting Your Performance Tuning for Threads

History
Historically, IBM i programmers have not had to worry very much about threads. True, they were introduced into the machine some time ago, but the average RPG application does not use them and perhaps never will, even if it is now allowed. Multiple-thread jobs have been fairly rare. That means that those who set up and organize IBM i subsystems (e.g. QBATCH, QINTER, MYOWNSUBSYSTEM, etc.) have not had to think much about the distinction between a "job" and a "thread."

Threads vs Jobs
But, threads are a good thing and so applications are increasingly using them. Especially for customers deploying (say) a significant new Java application, or Domino, a machine with the typical one-thread-per-job model may suddenly have dozens or even hundreds of threads in a particular job. Unfortunately, they are distinct ideas and certain AS/400 commands carefully distinguish them. If iSeries System Administrators are careless about these distinctions, as it is so easy to do today, poor performance can result as the system moves on to new applications such as Lotus Domino or especially Java.

With Java generally, and with certain applications, it will be commonplace to have multiple threads in a job. That means taking a closer look at some old friends: MAXACT and MAXJOB.

Recall that every subsystem has at least one pool entry. Recall further that, in the subsystem description itself, the pool number is an arbitrary number. What is more important is that the arbitrary number maps to a particular, real storage pool (*BASE, *SHRPOOL1, etc.). When a subsystem is actually started, the actual storage pool (*SHRPOOL1), if someone else isn't already using it, comes to life and obtains its storage.

However, storage pools are about more than storage. They are also about job and thread control. Each pool has an associated value called MAXACT that also comes into play. No matter how many subsystems share the pool, MAXACT limits the total number of threads able to reside and execute in the pool. Note that this is *threads* and not *jobs*.

Each subsystem, also, has a MAXJOBS value associated with it. If you reach that value, you are not supposed to be able to start any more jobs in the subsystem. Note that this is a *jobs* value and not a *threads* value. Further, within the subsystem, there are usually one or more JOBQs in the subsystem. Within each entry you can also control the number of jobs using a parameter. Due to an unfortunate turn in history, this parameter, which might more logically be called MAXJOBS today is called MAXACT. However, it controls *jobs*, not *threads*.

Problem
It is too easy to use the overall pool's value of MAXACT as a surrogate for controlling the number of Jobs. That is, you can forget the distinction between jobs and threads and use MAXACT to control the activity in a storage pool. But, you are not controlling jobs; you are controlling threads.

It is also too easy to have your existing MAXACT set too low if your existing QBATCH subsystem suddenly sees lots of new Java threads from new Java applications.

If you make this mistake (and it is easy to do), you'll see several possible symptoms:

- Mysterious failures in Java. If you set the value of MAXACT really low, certainly as low as one, sometimes Java won't run, but it also won't always give a graceful message explaining why.
- Mysterious "hangs" and slowdowns in the system. If you don't set the value pathologically low, but still too low, the system will function. But it will also dutifully "kick out" threads to a limbo known as "ineligible" because that's what MAXACT tells it to do. When MAXACT is too low, the result is useless wait states and a lot of system churn. In severe cases, it may be impossible to "load up" a CPU to a high utilization and/or response times will substantially increase.
- Note carefully that this can happen as a result of an upgrade. If you have just purchased a new machine and it runs slower instead of faster, it may be because you're using "yesterday's" limits for MAXACT

If you're having threads thrown into "ineligible", this will be visible via the WRKSYSSTS command. Simply bring it up, perhaps press PF11 a few times, and see if the Act->Inel is something other than zero. Note that other transitions, especially Act->Wait, are normal.

Solution
Make sure the *storage pool's* MAXACT is set high enough for each individual storage pool. A MAXACT of *NOMAX will sometimes work quite well, especially if you use MAXJOBS to control the amount of working coming into each subsystem.

Use CHGSHRPOOL to change the number of *threads* that can be active in the pool (note that multiple subsystems can share a pool):
CHGSHRPOOL ACTLVL(newmax)

Use MAXJOB in the subsystem to control the amount of outstanding work in terms of *jobs*:
CHGSBSD QBATCH MAXJOBS(newmax)

Use the Job Queue Entry in the subsystem to have even finer control of the number of jobs:
CHGJOBQE SBSD(QBATCH) JOBQ(QBATCH) MAXACT(newqueue job maximum)

Note in this particular case that MAXACT does refer to jobs and not threads.

19.2 General Performance Guidelines -- Effects of Compilation

In general, the higher the optimization, the less easy the code will be to debug. It may also be the case that the program will do things that are initially confusing.

In-lining

For instance, suppose that ILE Module A calls ILE Module B. ILE Module B is a C program that does allocation (malloc/free in C terms). However, in the right circumstances, compiler optimization will "inline" Module B. In-lining means that the code for B is not called, but it is copied into the calling module instead and then further optimized. So, for at least Module A, then, the "in-lined" Module B will cease to be an individual compiled unit and simply have its code copied, verbatim, into A.

Accordingly, when performance traces are run, the allocation activity of Module B will show up under Module A in the reports. Exceptions would also report the exception taking place in Module A of Program X.

In-lining of "final" methods is possible in Java as well, with similar implications.

Optimization Levels

Most of the compilers and Java support a reasonably compatible view of optimization. That is, if you specify OPTIMIZE(10) in one language, it performs similar levels of optimization in another language, including Java's CRTJVAPGM command. However, these things can differ at the detailed level. Consult the manuals in case of uncertainty.

Generally:

- OPTIMIZE(10) is the lowest and most debuggable.
- OPTIMIZE(20) is a trade-off between rapid compilation and some minimal optimization
- OPTIMIZE(30) provides a higher level of optimization, though it usually avoids the more aggressive options. This level can debug with difficulty.
- OPTIMIZE(40) provides the highest level of optimization. This includes sophisticated analysis, "code motion" (so that the execution results are what you asked for, but not on a statement-by-statement basis), and other optimizations that make debugging difficult. At this level of optimization, the programmer must pay stricter attention to the manuals. While it is surprisingly often irrelevant in actual cases, many languages have specific definitions that allow latitude to highly optimized compilers to do or, more importantly, "not do" certain functions. If the coder is not aware of this, the code may behave differently than expected at high optimization levels.

19.3 How to Design for Minimum Main Storage Use (especially with Java, C, C++)

The iSeries family has added popular languages whose usage continues to increase -- Java, C, C++. These languages frequently use a different kind of storage -- heap storage.

Many iSeries programmers, with a background in RPG or COBOL are unaware of the influence this may have on storage consumption. Why? Simply because these languages, by their nature, do not make much if any use of the heap. Meanwhile, C, C++, and Java very typically do.

The implications can be very profound. Many programmers are unclear about the tradeoffs and, when reducing memory usage, frequently attack the wrong problem. It is surprisingly easy, with these languages, to spend many megabytes and even hundreds of megabytes of main storage without really understanding how and why this was done.

Conversely, with the right understanding of heap storage, a programmer might be able to solve a much larger problem on the identical machine.

Theory -- and Practice
This is one place where theory really matters. Often, programmers wonder whether a theory applies in practice. After surveying a set of applications, we have concluded that the theory of memory usage applies very widely in practice.

In computer science theory, programmers are taught to think about how many "entities" there are, not how big the entity is. It turns out that controlling the number of entities matters most in terms of controlling main storage -- and even processor usage (it costs some CPU, after all, to *have* and *initialize* storage in the first place). This is largely a function of design, but also of storage layout. It is also knowing which storage is critical and which is not. Formally, the literature talks about:

Order(1) -- about one entity per system
Order(N) -- about "N" entities, where "N" are things like number of data base records, Java objects, and like items.
Order(N log N) -- this can arise because there is a data base and it has an accompanying index.
Order(N squared) -- data base joins of two data bases can produce this level of storage cost

Note the emphasis on "about." It is the number of entities in relation to the elements of the problem that count. An element of the problem is not a program or a subsystem description. Those are Order(1) costs. It is a data base record, objects allocated from the heap inside of loops, or anything like these examples. In practice, Order(N) storage predominates, so this paper will concentrate on Order(N).

Of course, one must eventually get down to actual sizes. Thus, one ends up with actual costs that get Order(N) estimated like this:

ActualCostForOrder(1) = a

ActualCostInBytes(N) = a + (b x N)

Where a and b are constants. "a" is determined by adding up things like the static storage taken up by the application program. "b" is the size of the data base record plus the size of anything else, such as a Java object, that is created one entity per data base record. In some applications, "N" will refer to some freestanding fact, like the maximum number of concurrent web serving operations or the number of outstanding new orders being processed.

However, the number of data base records will very often be the source of "N." Of course, with multiple data base files, there may be more than one actual "N". Still, it is usually true that the record count of one file compared to another will often be available as a ratio. For instance, one could have an "Order" record and average of three and a half "Order Detail" records. As long as the ratio is reasonably stable or can be planned at a stable value, it is a matter of convention which is picked to be "N" in that case; one merely adjusts "b" in the above equation to account for what is picked for "N".

System Level Considerations

In terms of the computer science textbooks, we are largely done. But, for someone in charge of commercial application deployment, there is one more practical thing to consider: Jobs and those newer items that now often come with them, threads.

Formally, if there is only one job or thread, then these are part of the Order(1) storage. If there are many, they end up proportional to N (e.g. One job for every 100,000 active records) and so are part of the Order(N) storage cost.

However, it is frequently possible to adjust these based on observed performance effects; the ratio to N is not entirely fixed. So, it is remains of interest to segregate these when planning storage. So, while they will not appear on the formal computer science literature, this paper will talk about Order(j) and Order(t) storage.

Typical Storage Costs

Here are typical things in modern systems and where they ordinarily sit in terms of their "entity" relationships.

Order(1)	Order(j)	Order(t)	Order(N)
ILE and OS/400 Programs	Just In Time compiled programs (Java *JIT)	Java threads	Data Base Records and IFS file records
Subsystem Descriptions	Total Job Storage	File Buffers of all kinds	Java (and C/C++) objects
Direct Execution Java Programs	Static storage from RPG and COBOL. Static final in Java.	SQL Result Set (nonrecord)	Operating System copies (e.g. Data Base) copies of application records
System values	Java Virtual Machine and most WebSphere storage	Program stack storage	SQL records in a result set

A Brief Example

To show these concepts, consider a simple example.

Part of a financial system has three logical elements to deal with:

1. An order record (order summary including customer information, sales tax, etc.)
2. An order detail record (individual purchased items, quantities, prices).
3. A table containing international currency rates of exchange between two arbitrary countries.

Question: What is more important? Reducing the cost of the detail record by a couple of bytes, or reducing the currency table from a cost of N squared (where "N" is the number of countries) to 2 times N.

There are two obvious implementations of the currency table:

1. Implement the table as a two dimensional array such that CurrencyExchangei,j will give the exchange between $country_i$ and $country_j$ for all countries.

2. Implement the table as a single dimension array with the *i*th element being the exchange rate between $country_i$ and the US dollar. One can convert to any country simply by converting twice; once to dollars and once to the other currency.

Clearly, the second is more storage efficient.

Now consider the first problem. The detail record looks like this:

Quantity as a four byte number (9B or 10B in RPG terms).
Name of the item (up to 60 characters)
Price of the item (as a zoned decimal field, 15 total digits with two decimal points).

A simple scrub would give:

Quantity as a two byte number (4B in RPG terms).
Name of the item (probably still 60 characters)
Price of the item (as a packed decimal field, probably 10 total digits with two decimal points).

How practical this change would be, if it represented a large, existing data base, would be a separate question. If this is at the initial design, however, this is an easy change to make.

Boundary considerations. In Java, we are done because Java will order the three entities such that the least amount of space is wasted. In C and C++, it might be possible to lay out the storage entities such that the compiler will not introduce padding between elements. In this particular example, the order given above would work out well.

Which is more important?

Reading the above superficially, one would expect the currency table improvement to matter most. There was a reduction from an N squared to an 2 times N relationship. However, this cannot be right. In fact, the number of countries is not "N" for this problem. "N" is the number of outstanding orders, a number that is likely in a practical system to be much larger than the number of countries. More critically, the

number of countries is essentially fixed. Yes, the number of countries in the world change from time to time. But, of course, this is not the same degree of change as order records in an order entry system. In fact, the currency table is part of the Order(1) storage. The choice between 2 times N and N squared should be based on whatever is operationally simpler.

Perform this test to know what "N" really is: If your department merged with a department of the same size, doing the same job, which storage requirements would double? It is these factors that reveal what the value of "N" is for your circumstances.

And, of course, the detail order record would be one such item. So, where are the savings? The above recommendations will save 9 bytes per record. If you write the code in RPG, this does not seem like much. That would be 9 bytes times the number of jobs used to process the incoming records. After all, there is only one copy of the record in a typical RPG program.

However, one must account for data base. Especially when accessing the records through an index of some kind, the number of records data base will keep laying about will be proportional to "N" -- the total number of outstanding orders. In Java, this can be even more clear-cut. In some Java programs, one processes records one at a time, just as in RPG. The most straightforward case is some sort of "search" for a particular record. In Java, this would look roughly the same as RPG and potentially consume the same storage.

However, Java can also use the power of the heap storage to build huge networks of records. A custom sort of some kind is one easy example of this.

In that case, it is easy for Java to contain the summary record and "dozens" of detail records, all at once, all connected together in a whole variety of ways. If necessary, modern applications might bring in the entire file for the custom sort function, which would then have a peak size at least as large as the data base file(s) itself or themselves.

Once you get above a couple hundred records, even in but one application, the storage savings for the record scrub will swamp the currency table savings. And, since one might have to buy for peak storage usage, even one application that references thousands of detail records would be enough to tip the scale.

A Short but Important Tip about Data Base

One thing easily misunderstood is variable length characters. At first, one would think every character field should be variable length, especially if one codes in Java, where variable length data is the norm.

However, when one considers the internals of data base, a field ought to be ten to twenty bytes long before variable length is even considered. The reason is, there is a cost of about ten bytes per record for the first variable length field. Obviously, this should not be introduced to "save" a few bytes of data.

Likewise, the "ALLOCATE" value should be understood (in OS/400 SQL, "ALLOCATE" represents the minimum amount of a variable record always present). Getting this right can improve performance. Getting it wrong simply wastes space. If in doubt, do not specify it at all.

A Final Thought About Memory and Competitiveness

The currency storage reduction example remains a good one -- just at the wrong level of granularity. Avoiding a SQL join that produces N squared records would be an example where the 2 times N alternative, if available, saves great amounts of storage.

But, more critically, deploying the least amount of Order(N) storage in actual implementation is a competitive advantage for your enterprise, large or small. Reducing the size of each N in main storage (or even on disk) eventually means more “things” in the same unit of storage. That is more competitive whether the cost of main storage falls by half tomorrow or not. More “things” per byte is always an advantage. It will always be cheaper. Your competitor, after all, will have access to the same costs. The question becomes: Who uses it better?

19.4 Memory Tuning Using the QPFRADJ System Value

The Performance Adjustment support (QPFRADJ system value) is used for initially sizing memory pools and managing them dynamically at run time. In addition, the CHGSHRPOOL and WRKSHRPOOL commands allow you to tailor memory tuning parameters used by QPFRADJ. You can specify your own faulting guidelines, storage pool priorities, and minimum/maximum size guidelines for each shared memory pool. This allows you the flexibility to set unique QPFRADJ parameters at the pool level.

For a detailed discussion of what changes are made by QPFRADJ, see the Work Management Guide. What follows is a description of some of the affects of this system value and some discussion of when the various settings might be appropriate.

When the system value is set to 1, adjustments are made to try to balance the machine pool, base pool, spooling pool, and interactive pool at IPL time. The machine pool is based on the amount of storage needed for the physical configuration of the system; the spool pool is fairly small and reflects the number of printers in the configuration. 70% of the remaining memory is allocated to the interactive pool; 30% to the base pool.

A QPFRADJ value of 1 ensures that memory is allocated on the system in a way that the system will perform adequately at IPL time. It does not allow for reaction to changes in workload over time. In general, this value is avoided unless a routine will be run shortly after an IPL that will make adjustments to the memory pools based on the workload.

When the system value is set to 2, adjustments are made as described, plus dynamic changes are made as changes in workload occur. In addition to the pools mentioned above, shared pools (*SHRPOOLxxx) are also managed dynamically. Adjustments are based on the number of jobs active in the subsystem using the pool, the faulting rates in the pool, and on changes in the workload over the course of time.

This is a good option for most environments. It attempts to balance system memory resources based on the workload that is being run at the time. When workload changes occur, such as time-of-day changes when one workload may increase while another may decrease, memory resources are gradually shifted to accommodate the heaviest loads.

When the system value is set to 3, adjustments are only made during the runtime, not as a result of an IPL.

This is a good option if you believe that your memory configuration was reasonable prior to scheduling an IPL. Overall, having the system value set to 2 or 3 will yield a similar effect for most environments.

When the system value is set to 0, no adjustments are made. This is a good option if you plan on managing the memory by yourself. Examples of this may be if you know times when abrupt changes in memory are likely to be required (such as a difference between daytime operations and nighttime operations) or when you want to always have memory available for specific, potentially sporadic work, even at the expense of not having that memory available for other work. It should be noted, however, that this latter case can also be covered by using a private memory pool for this work. The QPFRADJ system value only affects tuning of system-supplied shared pools.

19.5 Additional Memory Tuning Techniques

Expert Cache

Normally, the system will treat all data that is brought into a memory pool in a uniform way. In a purely random environment, this may be the best option. However, there are often situations where some files are accessed more often than others or when some are accessed in blocks of information instead of randomly. In these situations, the use of "Expert Cache" may improve the efficiency of the memory in a pool. Expert Cache is enabled by changing the pool attribute from *FIXED to *CALC. One advantage for using Expert Cache (*CALC) is that the system dynamically determines which objects should have larger blocks of data brought into main storage. This is based on how frequently the object is accessed. If the object is no longer accessed heavily, the system automatically makes the storage available for other objects that are accessed. If the newly accessed objects then become heavily accessed, the objects have larger blocks of data placed in main storage.

Expert Cache is often the best solution for batch processing, when relatively few files may be accessed in large blocks at a time or in sequential order. It is also beneficial in many interactive environments when files of differing characteristics are being accessed. The pool attribute can be changed from *FIXED to *CALC and back at any time, so making a change and evaluating its affect over a period of time is a fairly safe experiment.

More information about Expert Cache can be found in the Work Management guide.

In some situations, you may find that you can achieve better memory utilization by defining the caching characteristics yourself, rather than relying on the system algorithms. This can be done using the QWCCHGTN (Change Pool Tuning Information) API, which is described in the Work Management API reference manual. This API was provided prior to the offering of the *CALC option for the system. It is still available for use, although most situations will see relatively little improvement over the *CALC option and it is quite possible to achieve less improvement than with *CALC. When the API is used to adjust the pool attribute, the value that is shown for the pool is USRDFN (user defined).

SETOBJACC (Set Object Access)

In some cases, the object access performance is improved when the user manually defines (names a specific object) which object is placed into main storage. This can be achieved with the SETOBJACC command. This command will clear any pages of an object that are in other storage pools and moves the object to the specified pool. If the object is larger than the pool, the first portions of the object are replaced with the later pages that are moved into the pool. The command reports on the current amount of storage that is used in the pool.

If SETOBJACC is used when the QPFRADJ system value is set to either 2 or 3, the pool that is used to hold the object should be a private pool so that the dynamic adjustment algorithms do not shrink the pool because of the lack of job activity in the pool.

Large Memory Systems

Normally, you will use memory pools to separate specific sets of work, leaving all jobs which do a similar activity in the same memory pool. With today's ability to configure many gigabytes of mainstore, you may also find that work can be done more efficiently if you divide large groups of similar jobs into separate memory pools. This may allow for more efficient operation of the algorithms which need to search the pool for the best candidates to purge when new data is being brought in. Laboratory experiments using the I/O intensive CPW workload on a fully configured 24-way system have shown about a 2% improvement in CPU utilization when the transaction jobs were split among pools of about 16GB each, rather than all running in a single memory pool.

19.6 User Pool Faulting Guidelines

Each customer needs to track response time, throughput, and cpu utilization against the paging rates to determine a reasonable paging rate.

There are two choices for tuning user pools:

1. Set system value QPFRADJ = 2 or 3, as described earlier in this chapter.
2. Manual tuning. Move storage around until the response times and throughputs are acceptable. The rest of this section deals with how to determine these acceptable levels.

To determine a reasonable level of page faulting in user pools, determine how much the paging is affecting the interactive response time or batch throughput. These calculations will show the percentage of time spent doing page faults.

The following steps can be used: (all data can be gathered w/STRPFRMON and printed w/PRTSYSRPT). The following assumes interactive jobs are running in their own pool, and batch jobs are running in their own pool.

Interactive:

1. flts = sum of database and non-database faults per second during a meaningful sample interval for the interactive pool.

2. rt = interactive response time for that interval.

3. diskRt = average disk response time for that interval.

4. tp = interactive throughput for that interval in transactions per second. (transactions per hour/3600 seconds per hour)

5. fltRtTran = diskRt * flts / tp = average page faulting time per transaction.

6. flt% = fltRtTran / rt * 100 = percentage of response time due to

7. If flt% is less than 10% of the total response time, then there's not much potential benefit of adding storage to this interactive pool. But if flt% is 25% or more of the total response time, then adding storage to the interactive pool may be beneficial (see NOTE below).

Batch:

1. flts = sum of database and non-database faults per second during a meaningful sample interval for the batch pool.

2. flt% = flts * diskRt X 100 = percentage of time spent page faulting in the batch pool. If multiple batch jobs are running concurrently, you will need to divide flt% by the number of concurrently running batch jobs.

3. batchcpu% = batch cpu utilization for the sample interval. If higher priority jobs (other than the batch jobs in the pool you are analyzing) are consuming a high percentage of the processor time, then flt% will always be low. This means adding storage won't help much, but only because most of the batch time is spent waiting for the processor. To eliminate this factor, divide flt% by the sum of flt% and batchcpu%. That is: **newflt% = flt% / (flt% + batchcpu%)**
   This is the percentage of time the job is spent page faulting compared to the time it spends at the processor.

4. Again, the potential gain of adding storage to the pool needs to be evaluated. If flt% is less than 10%, then the potential gain is low. If flt% is greater than 25% then the potential gain is high enough to warrant moving main storage into this batch pool.

NOTE:
It is very difficult to predict the improvement of adding storage to a pool, even if the potential gain calculated above is high. There may be instances where adding storage may not improve anything because of the application design. For these circumstances, changes to the application design may be necessary.

Also, these calculations are of limited value for pools that have expert cache turned on. Expert cache can reduce I/Os given more main storage, but those I/Os may or may not be page faults.

19.7 POWER6 520 Memory Considerations

Because of the design of the POWER6 520 system, there are some key factors with the memory subsystem that one should keep in mind when sizing this system. The Power6 520, unlike the Power6

570, has no L3 cache, which does have an effect on memory sensitive workloads, like Java applications for instance. Having no L3 cache makes memory speed, or the bandwidth rating in megabytes per second, even more critical for memory sensitive workloads. The Power6 520 has 8 memory DIMM slots, which are positioned in groups of four behind each of the Power6-SCM modules and each group of four will be referred to as a quad for this discussion. The available number of active memory slots depends on the Processor Feature Code of the system.

When only one SCM module is installed, only one quad of memory is active and all slots must contain DIMMs of the same size and speed. When two SCM modules are installed (except in the case of the 4-way capable Capacity-on-Demand model with only one module enabled, which activates both memory quads), both quads of memory are active. When both are active, it is important to note that the first and second modules are separate and independent. So this means that even though the size and speed of memory DIMMs behind each module have to be the same, the size and speed of memory DIMMs behind the first module do not have to match the memory DIMMs behind the second module. For DIMMs ranging from 512 MB to 4 GB, the speed is 667 Mbps (PC2-5300). The 8 GB DIMMs are different however, with a speed of 400 Mbps (PC2-3200). This decrease in speed for 8 GB DIMMs can have a negative effect on performance with memory sensitive workloads. This effect, along with the fact that there is no L3 cache, should be considered when planning for current and future growth and also LPAR configurations.

To test the performance difference of 4 GB DIMMs versus 8 GB DIMMs (essentially testing the difference in speed) and what occurs when the DIMMs of different sizes are "mixed", we used a Power6 520 (9408-M25) F/C 5635 (a fully enabled system) with one partition using all the available resources. "Mixed" here means the DIMMs in one quad behind a module are 4 GB and the DIMMs in the opposite quad are 8 GB. We started with a baseline consisting of all 4 GB DIMMs behind both modules, which is the best performing case. Then switched to all 8 GB DIMMs behind both modules and ran the same tests again. The performance of the workloads that were memory sensitive followed suit with the decrease in memory speed, which was expected. This is very important to consider when considering the amount of memory needed for a system. Deciding to go with the larger capacity 8 GB DIMMs does reduce your memory's speed and can have a negative performance effect on your workload. Of course each workload will behave differently based on its sensitivity to memory.

Next we placed 4 GB DIMMs behind one module and 8 GB DIMMs behind the opposite module. Because the one module had the faster 4 GB DIMMs behind it, the same workloads produced results that ranged between the best case, all 4 GB DIMMs, and the worst case, all 8 GB DIMMs. Again, we used only one partition that utilized all the available resources, but there are other factors to consider when using LPAR.

LPAR, or Logical Partitioning, increases flexibility, enabling selected system resources like processors, memory and I/O components to be utilized by various partitions, either in a shared or dedicated environment, on the same system. In the "mixed" environment previously described, it is possible to have one partition utilizing memory on 4 GB DIMMs and a second partition, configured with exactly the same amount of resources, utilizing memory on 8 GB DIMMs. This can cause an application to have different performance characteristics on the partitions. It is also possible for partitions to be assigned a mix of memory from different DIMMs, depending on how the memory is allocated at partition activation time. This means that a partition that requires 4 GB of memory could be assigned 2 GB from the quad with 4 GB DIMMs and the other 2 GB from the quad with 8 GB DIMMs. This too can cause an application to have different performance characteristics on partitions configured with exactly the same amount of resources.

When system planning for the Power6 520, there are a number of memory related factors that should be considered, each of which can affect performance of memory sensitive workloads. First and foremost, the Power6 520 has no L3 cache. Having no L3 cache makes memory speed even more critical for memory sensitive workloads. If memory capacity needs can be achieved with 4 GB DIMMs or smaller, this will give the best memory speed. If memory capacity needs result in mixing 4 GB and 8 GB DIMMs, that option is available, but can have a negative performance effect on memory sensitive workloads. Mixing DIMMs can also cause partitions configured with exactly the same amount of resources to have varying performance characteristics. Since the Power6 520 only has 8 available memory DIMM slots, memory capacity can be an issue. If memory capacity is a concern, the 8 GB DIMMs will increase the capacity, but result in a slower memory speed.

19.8 Aligning Floating Point Data on POWER6

The PowerPC architecture specifies that storage operands ought to be appropriately aligned. In many cases, there is a slight performance benefit and the compiler knows this. In other cases, the operands must be aligned for functional reasons. For example:

1. Pointers used by IBM i must be aligned on a 16-byte boundary,
2. PowerPC instructions in a program must be word aligned,
3. Binary Floating-Point operands ought to be word-aligned and should not cross a page boundary.

Other operand types allow generally free alignment of the data.

Although such a specification exists for Binary Floating-Point operands, the processor designs have the option of allowing free alignment of Binary Floating-Pointer operands as well. The POWER6 processors, however, took a different approach. If either a 4-byte short form or 8-byte long form are not word-aligned, the POWER6 processor will produce an alignment interrupt. Fortunately, the IBM i alignment interrupt handler recognizes this and does allow programs to successfully execute even if the Binary Floating-Point operand is not word aligned. However, this emulation of each such operation comes at a very considerable impact to the performance of such floating-point load and store instructions. While an appropriately aligned floating-point load or store can execute extremely rapidly, the emulation when misaligned can take thousands of times longer. If such accesses are rare compared to the remainder of the function being provided, this emulation may not matter to the performance of the application. As such floating-point accesses become more frequent, this emulation alone can account for most of the time spent within an application.

The compiler does attempt to assure that such Binary Floating-Point operands are at least word aligned. However, there are ways that the compiler's intent can be over-ridden. Packing data which includes floating-point variables within a structure may result in this occurring; packing of structures can occasionally save some space in memory. For this reason, it is prudent to assure that floating-point variables are allowed to be at least word aligned. If this can not be done, it may be appropriate to first copy the floating-point variables to a local aligned variable in storage; this may need to be done via an explicit move operation which is unaware of the type of the data for if the type is known. Without this the floating-point data may be copied using the floating-point loads and store, resulting in an alignment interrupt.

As an example, consider the following C-like structures, one specifying "packed" and the other allowed to be aligned per the compiler. For example:


```
struct FPAlignmentStruct Packed
{
        double FloatingPointOp1;
        char ACharacter;
        double FloatingPointOp2;        // Byte aligned; Can result in alignment interrupt.
}

struct FPAlignmentStructNormal          // Allows for preferred alignment
{
        double FloatingPointOp1;
        char ACharacter;
        double FloatingPointOp2;        // Compiler padding added.
}
```

The first of these structures uses packing in order to minimize the amount of storage used. Here the structure consumes exactly 17 bytes, 8 each for the two floating-point values and one byte for the character. Assuming that the first is doubleword aligned as preferred, the second floating-point variable will be aligned on a doubleword+1 boundary. Each access of this second floating-point variable will result in an interrupt on POWER6 processors.

The second of these structures allows the compiler to assure preferred alignment, but it consumes more storage (i.e., 24 bytes). The extra 7 bytes over the first comes from the compiler adding padding after the character variable in order to assure that the second floating-point variable is doubleword aligned.

If minimal storage is nonetheless required, there is another technique which will assure preferred alignment and minimal storage. This is accomplished by packaging the larger variables first as in the following example:

```
struct FPAlignmentStructNormal
{
        double FloatingPointOp1;
        double FloatingPointOp2;        // Aligned without padding.
        char ACharacter;
}
```

This structure is also seventeen bytes in size and does assure preferred alignment.

19.9 Energy Management

This section will briefly touch on some of the available options for managing energy usage and how it relates to performance. First thing to consider when deciding on an energy management option is if you are willing to trade off some amount of performance and/or capacity to save on energy consumption. **IBM Systems Director Active Energy Manager (AEM)** provides the ability to adjust energy consumption to different levels. Three Power States are available.

- **Static Power Save (SPS)** – This option will drop voltage and frequency of the cores.

- **Dynamic Power Save (DPS)** – This option will optimize power versus performance using dynamic voltage and frequency slewing. The frequency of the cores will never surpass nominal.
- **Dynamic Power Save – Favor Performance (DPS-FP)** – This option saves power in low utilization states, but will increase frequency as utilization increases. The frequency can reach a state above nominal when necessary.

Under the covers, the OS and the Hypervisor are also taking some steps to minimize power consumption. The OS will return any unused cores in the partition to the Hypervisor for napping. Napping is a function that allows the core to quiesce to a lower power state, leaving the cores cache active. Sleeping is the next level, which allows the full core to quiesce to a lower power state. Bringing a core out of the sleep state is fairly rapid, but it is slow enough that the hypervisor won't send a core to sleep mode unless it has been napping for some period of time.

19.10 Simultaneous Multi-Threading (SMT)

A technology that has been around for quite a few years now - called Simultaneous Multi-Threading (SMT) - has been enhanced within the POWER7 cores. Where preceding designs supported up to two threads of instruction streams per core, POWER7's design allows this to be expanded to four. Because of SMT4, each core provides still more performance capacity than its predecessors. Where cores are normally thought of as executing the instruction stream of individual tasks, SMT and POWER7's SMT4 provide the means of having each core concurrently execute the instruction streams of one to four tasks.

19.11 Power 780 TurboCore

The Power 780 model has the ability to switch between its standard MaxCore mode and TurboCore mode, where performance per core is boosted with access to both additional cache and additional clock speed. Based on the user's configuration option, any Power 780 system can be booted in standard mode, enabling up to a maximum of 64 processor cores running at 3.86 GHz, or in TurboCore mode, enabling up to 32 processor cores running at 4.14 GHz and twice the cache per core.

TurboCore option allows customers to trade system capacity for higher core performance in most customer environments through 7% additional frequency and 16MB of shared L3.1 cache. As partition size grows beyond drawer boundaries, the benefits of higher frequency and 16 MB shared L3.1 cache may have lesser impact in some workload environments.

Internal testing by IBM has shown as much as a 4-5% performance gain in CPW while ranging to over 15% in Java workloads given appropriate system configurations. Additional testing continues to quantify these results and will be posted as they become available. In the meantime, clients are urged to use the Workload Estimator Tool for sizing and consider the following factors as input to the result. Websphere and Java workloads, multiple smaller partition sizes, frequency sensitivity, and single threading tend to favor the TurboCore model while large partition sizes running large commercial database workloads or heavy I/O activity tend to favor the MaxCore model.

Chapter 20. IBM Systems Workload Estimator and IBM Systems Energy Estimator

20.1 Overview

The IBM Systems Workload Estimator and the IBM Systems Energy Estimator are integrated tools to assist you with performance sizing and energy estimation. Please refer to the sections below for more detailed information on each of these tools.

20.2 IBM Systems Workload Estimator

The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE), located at: http://www.ibm.com/systems/support/tools/estimator, is a web-based sizing tool for IBM Power Systems (including System i and System p) and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. The Workload Estimator allows measurement input to best reflect your current workload and provides a variety of built-in workloads to reflect your emerging application requirements. Virtualization can be reflected in the sizing to yield a more robust solution, by using various types of partitioning and virtual I/O. The Workload Estimator will provide current and growth recommendations for processor, memory, and disk that satisfy the overall customer performance requirements.

The Estimator supports sizings dealing with multiple systems, multiple partitions, multiple operating systems, and multiple time intervals. The Estimator also provides the ability to easily do multiple sizings. These features can be coordinated by using the functions on the Workload Selection screen.

The Estimator will recommend the system model, processor, memory, and disk requirements that are necessary to handle the overall workload with reasonable performance expectations. To use the Estimator, you select one or more workloads and answer a few questions about each workload. Based on the answers, the Estimator generates a recommendation and shows the predicted CPU utilization of the recommended system in graphical format. The results can be viewed, printed, or saved as a PDF. The visualize solution function can be used to better understand the recommendation in terms of time intervals and virtualization. The Estimator can also be optionally linked to the System Planning Tool so that the configuration and validation may continue.

Sizing recommendations from the Estimator are based on processing capacity, which reflect the system's overall ability to handle the aggregate transaction rate. Again, this recommendation will yield processor, memory, and disk requirements. Other aspects of sizing must also be considered beyond the scope of this tool. For example, to satisfy overnight batch windows or to deal with single-threaded applications, there may be additional unique hardware requirements that would allow adequate completion time. Also, you may need to increase the overall disk recommendation to ensure that there is enough space to satisfy the overall storage requirements.

Sizing recommendations start with benchmarks and performance measurements based on well-defined, consistent workloads. For the built-in workloads in the Estimator, measurements have been done with numerous systems to characterize the workloads. Most of those workloads have parameters that allow them to be tailored to best suit the customer environment. This, again, is based on measurements and feedback from customers and Business Partners. Keep in mind, however, that many of these technologies

are constantly evolving. IBM will continue to refine these workloads and sizing rules of thumb as IBM and our customers gain more experience.

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 only.

Caveats and disclaimers such as the paragraph above are frequently used with regard to performance and estimating performance. The following two paragraphs are statements that the Estimator includes with every estimate and are provided here for reference. Please note in particular the comments in the second paragraph below having to do with **multi-threading** and **single-threaded applications**.

Note: The 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).

Note: 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.*

20.3 Merging PM for Power Systems data into the Workload Estimator

The Workload Estimator is designed to accept data from various sources including the Performance Management (PM) for Power Systems tool. PM for Power Systems includes the former PM for System i™ and PM for System p™ tools. PM for Power Systems assists with many of the functions associated with capacity planning and performance analysis -- automatically. It will collect various data from your system that is critical to sizing and growth estimation. This data is then consolidated and sent to the Estimator. The Estimator will then use weekly statistics recorded from your system and show the system performance over time. The Estimator can use this information to more accurately determine the growth trends of the system workload.

The PM data is easily merged into the Estimator while viewing your PM graphs on the web. To view your PM for Power Systems graphs on the web, go to http://www.ibm.com/systems/power/support/pm. Choose the 'View your online reports' button in the PM for Power Systems reports box on the right side of the page.

Follow these instructions to merge the PM for Power Systems data into the Estimator:

1. Enter your user id/password on the PM web site.

2. Select the systems or partitions that you wish to size.

Your PM data is then passed into the Estimator. If this is your first time using PM data with the Estimator, it is recommended that you take a few minutes to read the Measured Workload Integration tutorial, found on the help/tutorial tab in the Estimator.

Additional information on the PM for Power Systems tool is available at http://www.ibm.com/systems/power/support/pm.

20.4 Workload Estimator Updates and Access

The intent is to provide a new version of the IBM Systems Workload Estimator 3 to 4 times per year. With each new version a "What's New" section is included which describes the latest features that have been added. Be sure to check out the latest function and also the links to descriptions of new function for previous releases.

The IBM Systems Workload Estimator is available in two formats, on-line and as a download. The on-line version is the preferred way to access the Estimator. The download includes the Workload Estimator Developer toolkit. Both formats are described and available at http://www.ibm.com/systems/support/tools/estimator.

It is also highly recommended that there should be involvement of IBM Sales or IBM Business Partners before making any purchasing decisions based on the results obtained from the Estimator.

20.5 What the Workload Estimator is Not

The Estimator focuses on sizing based on capacity for processor, memory, and disk. The Estimator does not recommend network adapters, communications media, I/O adapters, or configuration topology. The Estimator is not a configurator nor a configuration validation tool. The Estimator does not take into account features like detailed journaling, resource locking, single-threaded applications, time-limited batch job windows, or poorly tuned environments.

The Estimator is a capacity sizing tool. Even though it does not represent actual transaction response times, it does adhere to the policy of giving recommendations that abide by generally accepted utilization thresholds. This means that the recommendation will likely have acceptable response times with the solution being prior to the knee of the curve on the common throughput vs. response time graph.

20.6 IBM Systems Workload Estimator Developer toolkit

The IBM Systems Workload Estimator Developer (a.k.a., the Developer or Workload Developer toolkit) is a downloadable tool that is used to create workloads for the IBM Systems Workload Estimator. Using the Developer, users create what is essentially an XML document which is comprised of several components including questions to prompt for input, algorithms to manipulate the input, and specifications for the WLE sizing engine to prioritize the allowable recommendations to be presented to the user as the proposed hardware solution.

IBM has developed numerous workloads that are integrated within the Workload Estimator such as Web Serving, WebSphere™ Application Server, WebSphere Commerce Suite, IBM WebSphere Portal, WebFacing, Lotus Domino Mail, Lotus Quickr, Traditional OLTP, Existing System, and Generic, just to name a few. While these integrated workloads cover a wide spectrum of application types, there is really no possible way for IBM to develop a specific workload for the Estimator for each of the thousands of applications that various Business Partners offer. To address this need, IBM designed the capability for the Business Partner to provide a workload definition to the Estimator. This capability is referred to as creating a sizing guide for the Estimator. Information on creating and hosting a sizing guide can be found at http://ibm.com/servers/sizing/tools.

Numerous sizing guides have been created by business partners and companies for specific products and applications. You can search for sizing guides at http://ibm.com/servers/sizing/sizingguides. To search for the publicly available sizing guides, select the "Public Web-based Sizing Guides" option from the drop-down for the first input field titled "Choose type of Sizing Guides". Then select additional options to narrow your search based on server platform, company name, or a solution name.

The Developer is included in the Workload Estimator download available at: http://www.ibm.com/systems/support/tools/estimator. To download, scroll to the bottom of the page to the section titled "Download version". Then click on "Download the IBM Systems Workload Estimator (includes Workload Developer Toolkit)."

20.7 IBM Systems Energy Estimator

The IBM Systems Energy Estimator (a.k.a., Energy Estimator) is a web-based tool for estimating power requirements for IBM Power Systems™ available at http://www.ibm.com/systems/support/tools/estimator/energy. You can use this tool to estimate typical power requirements (watts) for a specific system configuration under normal operating conditions. Systems currently supported by the Energy Estimator include IBM Power Systems using POWER6 and POWER7 processor technology, and IBM System i* and System p* models using POWER6 processor technology.

To create an energy estimate for a given system, you begin by choosing a processor model for which you'd like to generate an energy estimation. Once you've selected the processor model, you will see additional input fields appropriate to that system model for components such as memory DIMMs, PCI resources and internal media. After specifying the desired inputs, the tool provides an estimated energy value in watts and BTU/hr for the described system configuration. For the IBM Power 575 supercomputing node, which includes water-cooled components, the estimated energy value is also broken out by air and water contribution.

Additional capabilities of the Energy Estimator let you save a system configuration that you've defined and restore it later. A saved configuration may include processor model, number of processor cores, number and type of memory DIMMS, number of PCI resources, internal media resources, number of expansion disks, etc. The Download PDF option lets you create and download a PDF file which captures the defined system configuration, the energy estimate information in watts and BTU/hr, a list of the included components with feature codes, and a bibliography of energy-related articles and resources. Supportive help text is available on all screens to assist you in achieving the desired outcomes.

The IBM Systems Energy Estimator runs as a stand alone tool and is also integrated with the IBM Systems Workload Estimator. From the Workload Estimator's Selected system screen you can select the

green, Energy estimate option to generate estimated energy requirements for the Immediate and Growth solutions.

The Energy Estimator supports import of XML files exported from the IBM Configurator for e-business (e-config) tool. After import of an e-config XML file, the Energy Estimator provides an estimated energy value for the configuration described in the e-config file. The Energy Estimator recognizes processor activation codes specified in the e-config file and will adjust the estimated energy result for active versus inactive cores, based on available energy data within the Energy Estimator. Users are able to specify an expected CPU utilization for system configurations imported from e-config for models with utilization-based energy estimation capability.

The Energy Estimator is also integrated with the System Planning Tool (SPT). System configurations defined in SPT, also referred to as system plans, can be saved to files with an extension of .sysplan. SPT integration with the Energy Estimator enables .sysplan files created from SPT to be imported into the Energy Estimator. The Energy Estimator then provides an estimated energy value for the configuration described in the system plan and also lets you modify system components within the Energy Estimator to allow comparisons involving energy requirements. Within SPT, users will find new information available on the summary tab, which provides an estimated energy value for the defined system configuration.

An article in the March edition of IBM Systems Magazine featured the IBM Systems Energy Estimator and provides additional information about the tool. The article, titled “The Power of Prediction - Estimating energy requirements for IBM Power Systems”, is available at http://www.ibmsystemsmag.com/ibmi/march09/features/24362p1.aspx.

Appendix A. CPW Rating Description

"Due to road conditions and driving habits, your results may vary." "Every workload is different." These are two hallmark statements of measuring performance in two very different industries. They are both absolutely correct. For systems that run IBM i, IBM has provided a measure called the CPW rating to represent the relative computing power (more specifically, transactional capacity) of these systems in a commercial environment. The type of caveats listed above are always included because no prediction can be made that a specific workload will perform in the same way that the workload used to generate CPW information performs.

The CPW rating provides a measure to show how on-line transactions processing (OLTP) workloads perform on systems that run IBM i. The CPW rating is built using workloads that can utilize the full processing power of the system. This includes processor capabilities such as SMT (simultaneous multi-threading) and optionally enabled features such as TurboCore.

Many, but clearly not all, IBM i applications tend to follow the same patterns as the CPW rating - which stands for **Commercial Processing Workload rating**. These applications tend to have many jobs running brief transactions in an environment that is dominated by IBM system code performing database operations. The CPW rating is not intended to represent workloads that are single-threaded ("batch" jobs can be a subset of this class of applications). Single-threaded workloads tend to consume a single processor or processor thread for an extended period of time and utilize different CPU pathlengths and I/O characteristics from OLTP workloads. Therefore single-threaded workloads that are typically found in batch environments tend to have different characteristics than what is represented by the CPW rating. The CPW rating is also not intended to represent applications which spend a large portion of their overall processor pathlength in application code. These applications tend to have different scaling behaviors than the CPW rating due to longer pathlength per transaction and less I/O processing.

The CPW rating is a self-referential capacity metric. Because of this, it should be used for representing the relative capacity of different systems running IBM i. Such capacity metrics can not be used to represent the execution speed of any given thread of execution. Use the IBM Systems Workload Estimator sizing tool (see Appendix B for details) for assistance in sizing systems for specific workloads.

A.1 CPW Rating

The CPW rating of a system is generated 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.).

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 super-tuned 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

CPW Rating deployment

For systems that were announced before October 2011, the CPW3 workload (or it’s predecessor workloads) was used to characterize system performance. The results were provided as a CPW rating. Starting with the October 2011 system announcements, a new workload called “COPR” will be used to provide performance results that produce the CPW rating. This new workload will allow IBM to provide CPW rating information more effectively. The resulting CPW rating is very similar between the two workloads. For OLTP workload sizing, there should be virtually no difference between the previous CPW3-based CPW rating and the new COPR-based CPW rating.

There is no plan to publish new unique COPR-based metrics. This would be of little value without establishing measurements over a wide range of older servers for comparison points. Plus the similarity

of the COPR workload and the CPW3 workload metrics means that it would not be expected to change any decision making parameters.

A.2 CPW3 (Commercial Processing Workload)

The CPW3 workload simulates the database server of an OLTP environment. Requests for transactions are received from an outside source and are processed by application service jobs on the database server. It is based, in part, on the business model from benchmarks owned and managed by the Transaction Processing Performance Council. However, there are substantive differences between this workload and public benchmarks that preclude drawing any correlation between them. For more information on public benchmarks from the Transaction Processing Performance Council, refer to their web page at www.tpc.org.

There are five business functions of varying complexity that are simulated. These transactions are all executed by batch server jobs, although they could easily represent the type of transactions that might be done interactively in a customer environment. Each of the transactions interacts with 3-8 of the 9 database files that are defined for the workload. Database functions and file sizes vary. Functions exercised are single and multiple row retrieval, single and multiple row insert, single row update, single row delete, journal, and commitment control. These operations are executed against files that vary from 100's of rows to 100's of millions of rows. Some files have multiple indexes, some only one. Some accesses are to the actual data and some take advantage of advanced functions such as index-only access.

A.3 COPR (Commercial Performance Rating)

We are introducing a new OLTP workload called COPR (COmmercial Performance Rating). It's purpose and characteristics are very much like that of the CPW3 workload.

As with the CPW3 workload, COPR is a relative-performance workload, not a benchmark. Although roughly based upon a public benchmark, it is to be used to assist in determining the relative performance capacity of various commercial POWER based systems. It is not unduly optimized to produce the very best performance ratings - as would be the case in a benchmark - but instead uses capabilities expected to be used by customers. As the name COPR - Commercial Performance Rating - implies, its purpose is to provide guidance for gauging system capacity. Since it is an OLTP workload, the focus of COPR is on many jobs that run simultaneously and execute relatively short transactions, similar to the CPW3 workload concepts.

What are the reasons for migrating to a new workload to generate the CPW rating for IBM i environments? The CPW3 workload and its variations have been used as a relative performance workload for many years and will continue to be so. Over time, though, the means of and support for database operations have changed. Where the CPW3 workload is largely based upon languages like RPG and COBOL using native database interfaces, the COPR workload accesses the database tables using a higher level query language (e.g., SQL, JDBC) and stored procedures.

As with the CPW3 workload, COPR acts primarily as a database server with a set of jobs - "Job Sets" in COPR nomenclature - acting independently to drive the random high level database requests. The number of jobs accepting such input is set to exceed the number of "processors" (i.e., the number of

processor cores multiplied by the SMT - Simultaneous Multi-Threading - capability of each core) by enough to tend to keep all “processors” busy much of the time. This also means that the many database tables and indexes are frequently being concurrently accessed, strongly and intentionally driving database contention and integrity capabilities.

The types of transactions executed by COPR tend to be more complex and longer running than those found in the CPW3 workload. COPR spends much of its processing time doing what you would expect it to be doing, executing within the IBM i componentry supporting such database accesses.

The COPR workload allows IBM to be more effective in providing CPW rating information. The robust nature of the COPR workload also helps IBM better leverage performance insights for our operating system and firmware development teams.

Appendix B. IBM i Sizing and Performance Data Collection Tools

The following section presents some of the tools available for sizing and capacity planning. (Note: There are products from vendors not included here that perform similar functions.) All of the tools discussed here support the current range of System i products, and include the capability to model logical partitions, partial processors (micropartitions) and server workload consolidation.

- Performance Data Collection Services
  This tool which is part of the operating system collects system and job performance data which is the input for many of the performance tools that are available today. Collection Services is started automatically when subsystem QSYSWRK is started.

  The default collection library is QPFRDATA but QMPGDATA may still be used if set up in a prior release. Collected data is stored in Management Collection Objects (type *MGTCOL). The CRTPFRDTA command is used to process that data and produce the performance database files used by other tools. CRTPFRDTA may be run manually or configured within collection services to run automatically during collection. For more information on Collection Services see the IBM i information center web page at
  http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzahx/rzahxcollectdatacs.htm

- IBM Systems Workload Estimator
  The IBM Systems Workload Estimator (a.k.a., the Estimator or WLE) is a web-based sizing tool for System i, System p, and System x. You can use this tool to size a new system, to size an upgrade to an existing system, or to size a consolidation of several systems. *See the IBM Systems Workload Estimator and IBM Systems Energy Estimator chapter for more information.*

- System i Batch Modeling tool STRBCHMDL. (BATCH400)
  This is best for MES upgrade sizing where the 'Batch Window' is important. BCHMDL uses Collection Services data to allow the user to view the batch jobs on a timeline. The elapsed time components (cpu, cpu queuing, disk, disk queuing, page faulting, etc.) are also available for viewing. The user can change the jobs or the configuration and run an analysis to determine the effect on batch runtime. The user can also model the effect of changing a single job into multiple jobs running concurrently. It can be found at: http://www.ibm.com/systems/i/advantages/perfmgmt/sizing.html

  Note: Modeling of batch workloads should only be done within a given server family, and should not be compared across server families such as POWER6 versus POWER7 due to differences in GHz ratings and Simultaneous Multi-Threading behaviors.

For more information on other System i Performance Tools, see the Performance Management web page at http://www.ibm.com/systems/power/software/i/management/performance/index.html and the IBM Redbook End to End Performance Management on IBM I SG24-7808-00
http://publib-b.boulder.ibm.com/abstracts/sg247808.html?Open

Appendix C. CPW Rating Relative Performance Values for IBM i

This chapter details the relative system performance values:

- Commercial Processing Workload (**CPW**). For a detailed description, refer to *Appendix A, "CPW Benchmark Description"*. CPW rating values are relative system performance metrics and reflect the relative system capacity for the OLTP workloads. CPW rating values can be used with caution in a capacity planning analysis (e.g., to scale CPU-constrained capacities, CPU time per transaction). However, these values may not appropriately reflect the performance of workloads than OLTP because of differing detailed characteristics (e.g., cache miss ratios, average cycles per instruction, software contention, I/O characteristics, memory requirements, and application performance characteristics). The CPW rating values shown in the tables are based on IBM internal tests. Actual performance in a customer environment may vary significantly. Use the "IBM Systems Workload Estimator" for assistance with sizing.

- Mail and Calendar Users (**MCU**). MCU values are no longer utilized or provided here.

- Compute Intensive Workload (**CIW**). CIW values are no longer utilized or provided here.

- User-based Licensing. Many newer models utilize user-based licensing for i5/OS. For assistance in determining the required number of user licenses, see the product web pages (for example: http://www.ibm.com/systems/i/hardware or http://www.ibm.com/systems/power/hardware) . Note that user-based licensing is not a performance statement or a replacement for system sizing; instead, user-based licensing only enables appropriate user connectivity to the system. Application environments differ in their requirements for system resources. Use the "IBM Systems Workload Estimator" for assistance with sizing based on performance.

- Relative Performance metric for System p (**rPerf**). System i systems that run AIX can be expected to produce the same performance as equivalent System p models given the same memory, disk, I/O, and workload configurations. The relative capacity of System p is often expressed in terms of rPerf values. The definition and the performance ratings for System p can be found at:
  - rPerf definition: http://www.ibm.com/systems/p/hardware/rperf.html
  - rPerf table: http://www.ibm.com/systems/p/hardware/system_perf.html

C.1 IBM i 7.1 Additions (February 2013)

C.1.1 POWER 710, 720, 730, and 740 models

This section provides CPW values for the POWER 710 models, POWER 720 models, POWER 730 models, and POWER 740 models announced in February 2013.

C.1.1.1 CPW values for IBM POWER Systems - IBM i operating system - model 710

Table C.1.1.1. *CPW values for IBM POWER System Model 710*

Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
710 (8231-E1D)	EPCE	3.6	4	28400
710 (8231-E1D)	EPCG	4.2	6	49400
710 (8231-E1D)	EPCJ	4.2	8	64500

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 socket configuration

C.1.1.2 CPW values for IBM POWER Systems - IBM i operating system - model 720

Table C.1.1.2. *CPW values for IBM POWER System Model 720*

Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
720 (8202-E4D)	EPCK	3.6	4	28400
720 (8202-E4D)	EPCL	3.6	6	42400
720 (8202-E4D)	EPCM	3.6	8	56300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 socket configuration

C.1.1.3 CPW values for IBM POWER Systems - IBM i operating system - model 730

Table C.1.1.3. *CPW values for IBM POWER System Model 730*

Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
730 (8231-E2D)	EPCF	4.3	8	59700
730 (8231-E2D)	EPCG	4.2	12	89200

Table C.1.1.3. CPW values for IBM POWER System Model 730				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
730 (8231-E2D)	EPCH	3.6	16	104700
730 (8231-E2D)	EPCJ	4.2	16	117600

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 2 socket configuration

C.1.1.4 CPW values for IBM POWER Systems - IBM i operating system - model 740

Table C.1.1.4. CPW values for IBM POWER System Model 740				
Model	Processor Feature	Chip Speed GHz	CPU Range(3)	Processor CPW
740 (8205-E6D)	EPCP	4.2	6-12	49000-91700
740 (8205-E6D)	EPCQ	3.6	8-16	56300-106500
740 (8205-E6D)	EPCR	4.2	8-16	64500-120000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The range of the number of processor cores per system.

C.1.2 POWER 750 models

This section provides CPW values for the POWER 750 models announced in February 2013.

C.1.2.1 CPW values for IBM POWER Systems - IBM i operating system - model 750

Table C.1.2.1. CPW values for IBM POWER System Model 750						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores
750 (8408-E8D)	EPT7	4.0	59000	108000	158000	208000
750 (8408-E8D)	EPT8	3.5	52000	96000	141500	185000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.1.3 POWER 760 models

This section provides CPW values for the POWER 760 models announced in February 2013.

C.1.3.1 CPW values for IBM POWER Systems - IBM i operating system - model 760

Table C.1.3.1. *CPW values for IBM POWER System Model 760*

			Processor CPW			
Model	Processor Feature	Chip Speed GHz	12 cores	24 cores	2x18 cores(3)	2x24 cores(4)
760 (9109-RMD)	EPT5	3.1	69800	129000	195700	258000
760 (9109-RMD)	EPT6	3.4	75200	137000	209000	274000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 36 core system was configured as 2 18-core partitions
3. The 48 core system was configured as 2 24-core partitions

C.2 IBM i 7.1 Additions (November 2012)

New POWER7+ based Compute Nodes for the IBM PureFlex System were announced in November 2012.

- IBM Flex System p260 compute node (7895-23X)

C.2.1 IBM Flex System p260

This section provides CPW values for the IBM Flex System p260 compute nodes announced in November 2012.

C.2.1.1 CPW values for IBM Flex System p260 compute nodes

Table C.2.1. *CPW values for IBM Flex System p260 compute nodes*

Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
p260 (7895-23X)	EPRD	4.0(5)	8 (3)	51400
p260 (7895-23X)	EPRB	3.6	16(4)	99500
p260 (7895-23X)	EPRA	4.1	16(4)	110000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. CPW value is for a 7-core partition with dedicated processors and a 1-core VIOS partition
4. CPW value is for a 15-core partition with dedicated processors and a 1-core VIOS partition
5. This model has 4 cores per socket; all others have 8 cores per socket

C.3 IBM i 7.1 Additions (October 2012)

New POWER7+ system models were announced in October 2012.
- 9117-MMD
- 9179-MHD

C.3.1 IBM POWER 770 and 780 models

This section provides CPW values for the POWER 770 and 780 models announced in October 2012.

C.3.1.1 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature EPM0

Table C.2.1.1. *CPW values for IBM POWER System Models*							
			Processor CPW				
Model	Processor Feature	Chip Speed GHz	6 cores [3]	9 cores	12 cores	24 cores	2x24 cores[4]
770 (9117-MMD)	EPM0	4.22	45800	68200	90000	154800	306600

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. This 770 processor feature EPM0 has 3-cores per socket.
4. The 48 core system was configured as 2 24-core partitions

C.3.1.2 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature EPM1

Table C.3.1.2. *CPW values for IBM POWER System Models*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	16 cores	32 cores	2x24 cores[3]	2x32 cores[4]
770 (9117-MMD)	EPM1	3.80	28700	56100	110000	191500	290500	379300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24-core partitions
4. The 64 core system was configured as 2 32-core partitions

C.3.1.3 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature EPH0

Table C.3.1.3. *CPW values for IBM POWER System Models*	
	Processor CPW

Table C.3.1.3. CPW values for IBM POWER System Models								
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	16 cores	32 cores	2x24 cores(3)	2x32 cores(4)
780 (9179-MHD)	EPH0	4.42	32400	63200	123500	214000	326100	424400

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24-core partitions
4. The 64 core system was configured as 2 32-core partitions

C.3.1.4 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature EPH2

Table C.3.1.4. CPW values for IBM POWER System Models								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	32 cores	2x32 cores(3)	3x32 cores(4)	4x32 cores(5)
780 (9179-MHD)	EPH2	3.72	56000	108500	209500	414900	622300	829800

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 64 core system was configured as 2 32-core partitions
4. The 96 core system was configured as 3 32-core partitions
5. The 128 core system was configured as 4 32-core partitions

C.4 IBM i 7.1 Additions (April 2012)

New POWER7 based Compute Nodes for the IBM PureFlex System were announced in April 2012.
- IBM Flex System p260 compute node (7895-22X)
- IBM Flex System p460 compute node (7895-42X)

C.4.1 IBM Flex System p260 and p460

This section provides CPW values for the IBM Flex System p260 and p460 compute nodes announced in April 2012.

C.4.1.1 CPW values for IBM Flex System p260 and p460 compute nodes

Table C.4.1. *CPW values for IBM Flex System p260 and p460 compute nodes*				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
p260 (7895-22X)	EPR1	3.3[6]	8 [3]	38500
p260 (7895-22X)	EPR3	3.2	16[4]	80500
p260 (7895-22X)	EPR5	3.55	16[4]	87000
p460 (7895-42X)	EPR2	3.3[6]	16 [4]	80500
p460 (7895-42X)	EPR4	3.2	32 [5]	150000
p460 (7895-42X)	EPR6	3.55	32 [5]	162000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. CPW value is for a 7-core partition with dedicated processors and a 1-core VIOS partition
4. CPW value is for a 15-core partition with dedicated processors and a 1-core VIOS partition
5. CPW value is for a 30-core partition with dedicated processors and a 2-core VIOS partition
6. These models have 4 cores per socket; all others have 8 cores per socket

C.5 IBM i 7.1 Additions (October 2011)

New POWER7 system models were announced in October 2011.

- 9117-MMC
- 9179-MHC
- 8231-E1C & 8231-E2C
- 8202-E4C
- 8205-E6C

C.5.1 POWER 770 and 780 models

This section provides CPW values for the POWER 770 and 780 models announced in October 2011.

C.5.1.1 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature 4984

Table C.5.1.1. *CPW values for IBM POWER System Models*							
			Processor CPW				
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores	2x32 cores[3]
770 (9117-MMC)	4984	3.3	48200	93000	124400	162000	321100

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 64 core system was configured as 2 32-core partitions

C.5.1.2 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature 4983

Table C.5.1.2. *CPW values for IBM POWER System Models*							
			Processor CPW				
Model	Processor Feature	Chip Speed GHz	6 cores	12 cores	18 cores	24 cores	2x24 cores[3]
770 (9117-MMC)	4983	3.72	39800	77000	107500	135900	270500

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24-core partitions

C.5.1.3 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature 5003 with MaxCore mode

Table C.5.1.3. *CPW values for IBM POWER System Models*							
			Processor CPW				
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores	2x32 cores[4]
780 (9179-MHC)	5003	3.92	55200	106000	140700	183000	363000

*Note: 1. This processor feature is also available as a 4-core per chip configuration
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.
4. The 64 core system was configured as 2 32-core partitions

C.5.1.4 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature 5003 with TurboCore mode

Table C.5.1.4. *CPW values for IBM POWER System Models*				
Model	Processor Feature	Chip Speed GHz	Cores [4]	Processor CPW
780 (9179-MHC)	5003	4.14	1x8 cores	57450
780 (9179-MHC)	5003	4.14	2x8 cores	114850
780 (9179-MHC)	5003	4.14	3x8 cores	172450
780 (9179-MHC)	5003	4.14	4x8 cores	229650

*Note: 1. This processor feature is also available as a 8-core per chip configuration
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.
4. Each system was configured with partitions each of which are allocated with 8 processor cores

C.5.1.5 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature 4982 andMaxCore mode

Table C.5.1.5. *CPW values for IBM POWER System Models*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	6 cores	12 cores	24 cores	2x24 cores(4)	3x24 cores(5)	4x24 cores(6)
780 (9117-MHC)	EP24	3.44	36300	71400	138500	276000	413000	550700

*Note: 1. This processor feature is also available as a 4-core per chip configuration
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.
4. The 48 core system was configured as 2 24-core partitions
5. The 64 core system was configured as 3 24-core partitions
6. The 96 core system was configured as 4 24-core partitions

C.5.2 POWER 710, 720, 730, and 740 models

This section provides CPW values for the POWER 710 models, POWER 720 models, POWER 730 models, and POWER 740 models announced in October 2011.

C.5.2.1 CPW values for IBM POWER Systems - IBM i operating system - model 710

Table C.5.2.1. *CPW values for IBM POWER System Model 710*				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
710 (8231-E1C)	EPC1	3.0	4	23800
710 (8231-E1C)	EPC2	3.7	6	40900 (4)
710 (8231-E1C)	EPC3	3.55	8	51800 (4)

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 chip configuration
4. The value listed is unconstrained CPW (assuming that there is sufficient memory such that the processors would be the first constrained resource).

C.5.2.2 CPW values for IBM POWER Systems - IBM i operating system - model 720

Table C.5.2.2. *CPW values for IBM POWER System Model 720*				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
720 (8202-E4C)	EPC5	3.0	4	23800
720 (8202-E4C)	EPC6	3.0	6	34900
720 (8202-E4C)	EPC7	3.0	8	46300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 chip configuration

C.5.2.3 CPW values for IBM POWER Systems - IBM i operating system - model 730

Table C.5.2.3. *CPW values for IBM POWER System Model 730*				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
730 (8231-E2C)	EPC1	3.0	8	44600
730 (8231-E2C)	EPC4	3.7	8	51900
730 (8231-E2C)	EPC2	3.7	12	77200[(4)]
730 (8231-E2C)	EPC3	3.55	16	97700[(4)]

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 2 chip configuration
4. The value listed is unconstrained CPW (assuming that there is sufficient memory such that the processors would be the first constrained resource).

C.5.2.4 CPW values for IBM POWER Systems - IBM i operating system - model 740

Table C.5.2.4. *CPW values for IBM POWER System Model 740*				
Model	Processor Feature	Chip Speed GHz	CPU Range[(3)]	Processor CPW
740 (8205-E6C)	EPC9	3.3	4-8	25500-47800
740 (8205-E6C)	EPC8	3.7	4-8	27900-52200
740 (8205-E6C)	EPCA	3.7	6-12	41600-77200
740 (8205-E6C)	EPCB	3.55	8-16	52600-97700

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The range of the number of processor cores per system.

C.6 IBM i 7.1 Additions (April 2011)

This section provides CPW values for the POWER 750 models and the PS703/PS704 models announced in April 2011.

C.6.1 CPW values for IBM POWER Systems - IBM i operating system - model 750 features EPA1/EPA4

Table C.5.1. *CPW values for IBM POWER System Models*

			Processor CPW			
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores
750 (8233-E8B)	EPA4	3.2	47800	89600	131500	171400
750 (8233-E8B)	EPA1	3.6	52700	97000	141400	183200

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.6.2 CPW values for IBM POWER Systems - IBM i operating system - model 750 features EPA3

Table C.6.2. *CPW values for IBM POWER System Models*

			Processor CPW			
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	12 cores	16 cores
750 (8233-E8B)	EPA3	3.7	27300	51000	74700	97700

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.6.3 CPW values for IBM POWER Systems - IBM i operating system - model 750 features EPA2

Table C.6.3. *CPW values for IBM POWER System Models*

			Processor CPW			
Model	Processor Feature	Chip Speed GHz	6 cores	12 cores	18 cores	24 cores
750 (8233-E8B)	EPA2	3.7	40800	75500	109100	145600

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.6.4 CPW values for IBM POWER Systems - IBM i operating system - PS703/PS704 family

Table C.6.4. CPW values for IBM POWER System Models				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
PS703 (7891-73X)	52CC	2.4	16 (3)	64000 (5)
PS704 (7891-74X)	52CC	2.4	32 (4)	110000 (6)

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. CPW value is for a 15-core partition with dedicated processors and a 1-core VIOS partition
4. CPW value is for a 30-core partition with dedicated processors and a 2-core VIOS partition
5. The value listed is unconstrained CPW (assuming that there is sufficient memory such that the processor would be the first constrained resource).
6. The value listed is unconstrained CPW (assuming that there is sufficient disk I/O such that the processor would be the first constrained resource).

C.7 IBM i 7.1 Additions (August/October 2010)

This section provides CPW values for the POWER 710 models, POWER 720 models, POWER 730 models, POWER 740 models, POWER 750 models and the POWER 795 models announced in August 2010.

C.7.1 CPW values for IBM POWER Systems - IBM i operating system - model 710

Table C.7.1. CPW values for IBM POWER System Model 710				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
710 (8231-E2B)	8350	3.0	4	23800
710 (8231-E2B)	8349	3.7	6	40900 (4)
710 (8231-E2B)	8359	3.55	8	51800 (4)

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 chip configuration
4. The value listed is unconstrained CPW (assuming that there is sufficient memory such that the processors would be the first constrained resource).

C.7.2 CPW values for IBM POWER Systems - IBM i operating system - model 720

Table C.7.2. CPW values for IBM POWER System Model 720				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
720 (8202-E4B)	8350	3.0	4	23800
720 (8202-E4B)	8351	3.0	6	34900
720 (8202-E4B)	8352	3.0	8	46300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 1 chip configuration

C.7.3 CPW values for IBM POWER Systems - IBM i operating system - model 730

Table C.7.3. CPW values for IBM POWER System Model 730				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
730 (8231-E2B)	8350	3.0	8	44600
730 (8231-E2B)	8348	3.7	8	51900
730 (8231-E2B)	8349	3.7	12	77200(4)
730 (8231-E2B)	8359	3.55	16	97700(4)

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. Each model listed is a 2 chip configuration
4. The value listed is unconstrained CPW (assuming that there is sufficient memory such that the processors would be the first constrained resource).

C.7.4 CPW values for IBM POWER Systems - IBM i operating system - model 740

Table C.7.4. CPW values for IBM POWER System Model 740				
Model	Processor Feature	Chip Speed GHz	CPU Range(3)	Processor CPW
740 (8205-E6B)	8353	3.3	4-8	25500-47800
740 (8205-E6B)	8347	3.7	4-8	27900-52200
740 (8205-E6B)	8354	3.7	6-12	41600-77200
740 (8205-E6B)	8355	3.55	8-16	52600-97700

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The range of the number of processor cores per system.

C.7.5 CPW values for IBM POWER Systems - IBM i operating system - model 795 - feature 4702

Table C.7.5. *CPW values for IBM POWER System Model 795*						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	6 cores	12 cores	24 cores	48 (2x24 cores)[3]
795 (9119-FHB)	4702	3.7	39300	77600	149100	288500

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24-core partitions
4. The 795 model (feature 4702) can be configured as large as 192 cores total. Use IBM Systems Workload Estimator to configure systems larger than those listed in this document (http://www.ibm.com/systems/support/tools/estimator).

C.7.6 CPW values for IBM POWER Systems - IBM i operating system - model 795 - feature 4700

Table C.7.6. *CPW values for IBM POWER System Model 795*						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	32 cores	64 (2x32 cores)[3]
795 (9119-FHB)	4700	4.0	55100	107500	204300	399200

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 64 core system was configured as 2 32-core partitions
4. The 795 model (feature 4700) can be configured as large as 256 cores total. Use IBM Systems Workload Estimator to configure systems larger than those listed in this document (http://www.ibm.com/systems/support/tools/estimator).

C.7.7 CPW values for IBM POWER Systems - IBM i operating system - model 795 - feature 4700 with TurboCore mode

Table C.7.7. *CPW values for IBM POWER System Model 795 with TurboCore mode*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	12 cores	16 cores	24 cores	32 (2x16 cores)[3]
795 (9119-FHB)	4700	4.25	29300	59600	88800	115800	162100	218400

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 32 core system was configured as 2 16-core partitions
4. The 795 model (feature 4700) with TurboCore enabled can be configured as large as 128 cores total. Use IBM Systems Workload Estimator to configure systems larger than those listed in this document (http://www.ibm.com/systems/support/tools/estimator).

C.7.8 CPW values for IBM POWER Systems - IBM i operating system - model 750 feature 8336 (additional CPW values for IBM i 6.1.1)

Table C.7.8. *CPW values for IBM POWER System Models*						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores
750 (8233-E8B)	8336	3.55	52200	95700	138500	181000

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. All CPW values were measured with IBM i 6.1.1.

C.8 V6R1 Additions (April 2010)

This section provides CPW values for the IBM POWER systems announced in April 2010 and IBM POWER 780 TurboCore.

C.8.1 CPW values for IBM POWER Systems - IBM i operating system - PS700 family

Table C.8.1. *CPW values for IBM POWER System Models*				
Model	Processor Feature	Chip Speed GHz	CPUs	Processor CPW
PS700 (8406-70Y)	52CA	3.0	4 (3)	21100
PS701 (8406-71Y)	52C2	3.0	8 (4)	42100
PS702 (8406-71Y) + 8358	52C2	3.0	16 (5)	76300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. CPW value is for a 3.7-core partition with shared processors and a 0.3-core VIOS partition
4. CPW value is for a 7.5-core partition with shared processors and a 0.5-core VIOS partition
5. CPW value is for a 15-core partition with shared processors and a 1-core VIOS partition

C.8.2 CPW values for IBM POWER Systems - IBM i operating system - model 780 with TurboCore mode

Table C.8.2. *CPW values for IBM POWER System Models*				
Model	Processor Feature	Chip Speed GHz	Cores (4)	Processor CPW
780 (9179-MHB)	4982	4.14	1x8 cores	57450
780 (9179-MHB)	4982	4.14	2x8 cores	114850

Table C.8.2. CPW values for IBM POWER System Models				
Model	Processor Feature	Chip Speed GHz	Cores (4)	Processor CPW
780 (9179-MHB)	4982	4.14	3x8 cores	172450
780 (9179-MHB)	4982	4.14	4x8 cores	229650

*Note: 1. This processor feature is also available as a 8-core per chip configuration
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.
4. Each system was configured with partitions each of which are allocated with 8 processor cores

C.9 V6R1 Additions (February 2010)

This section provides CPW values for the POWER 750 models, POWER 770 models, and the POWER 780 models announced in February 2010. These models use POWER7 processor technology.

C.9.1 CPW values for IBM POWER Systems - IBM i operating system - model 750 features 8332/8334

Table C.9.1. CPW values for IBM POWER System Models						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	8 cores	16 cores	24 cores	32 cores
750 (8233-E8B)	8334	3.0	44600	82600	122500	158300
750 (8233-E8B)	8332	3.3	47800	88700	129700	168800

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.9.2 CPW values for IBM POWER Systems - IBM i operating system - model 750 feature 8335

Table C.9.2. CPW values for IBM POWER System Models						
			Processor CPW			
Model	Processor Feature	Chip Speed GHz	6 cores	12 cores	18 cores	24 cores
750 (8233-E8B)	8335	3.3	37200	69200	94900	135300

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.

C.9.3 CPW values for IBM POWER Systems - IBM i operating system - model 750 feature 8336

Table C.9.3. *CPW values for IBM POWER System Models*				
Model	Processor Feature	Chip Speed GHz	Cores	Processor CPW
750 (8233-E8B)	8336	3.55	32	181000

*Note: 1. This processsor feature is only available as a 32 core system
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.

C.9.4 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature 4981

Table C.9.4. *CPW values for IBM POWER System Models*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	16 cores	32 cores	2x24 cores(3)	2x32 cores(4)
770 (9117-MMB)	4981	3.1	22750	45000	88800	155850	229800	292700

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24-core partitions
4. The 64 core system was configured as 2 32-core partitions

C.9.5 CPW values for IBM POWER Systems - IBM i operating system - model 770 feature 4980

Table C.9.5. *CPW values for IBM POWER System Models*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	4 cores	6 cores	12 cores	18 cores	24 cores	2x24 cores(3)
770 (9117-MMB)	4980	3.5	24900	37400	73100	99000	131050	248550

*Note: 1. These configurations were run with SMT4 enabled
2. Nominal system values were used for energy settings.
3. The 48 core system was configured as 2 24core partitions

C.9.6 CPW values for IBM POWER Systems - IBM i operating system - model 780 feature 4982 and MaxCore mode

Table C.9.6. *CPW values for IBM POWER System Models*								
			Processor CPW					
Model	Processor Feature	Chip Speed GHz	4 cores	8 cores	16 cores	32 cores	2x24 cores(4)	2x32 cores(5)
780 (9179-MHB)	4982	3.86	26600	54400	105200	177400	265200	343050

*Note: 1. This processor feature is also available as a 4-core per chip configuration
2. These configurations were run with SMT4 enabled
3. Nominal system values were used for energy settings.
4. The 48 core system was configured as 2 24-core partitions
5. The 64 core system was configured as 2 32-core partitions

C.10 V6R1 Additions (April 2009)

C.10.1 CPW values for IBM Power Systems - IBM i operating system - model 520

Table C.10.1. *CPW values for Power System Models*

Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	CPUs	Processor CPW
520 (8203-E4A)	5577	4.7	2x4MB / 32MB	2	9500
520 (8203-E4A)	5587	4.7	2x4MB / 32MB	4	18300

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.

C.10.2 CPW values for IBM Power Systems - IBM i operating system - model 550

Table C.10.2. *CPW values for Power System Models*

				Processor CPW			
Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	2 cores	4 cores	6 cores	8 cores
550 (8204-E8A)	4967	5.0	2x4MB / 32MB	10600	20550	28800	37950

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.

C.10.3 IBM i5/OS running on IBM BladeCenter JS23/JS43 using POWER6 processor technology

Table C.10.3. *IBM BladeCenter models*

Blade Model	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	CPUs	Processor CPW
JS23 (7778-23X)	52C1	4200	2x4MB / 32 MB	3.7 of 4 (2)	14400
JS43 (7778-23X)	52C0	4200	2x4MB / 32 MB	7 of 8 (3)	24050

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. CPW value is for a 3.7-core partition with shared processors and a 0.3-core VIOS partition

3. CPW value is for a 7-core dedicated partition and a 1-core VIOS

C.11 V6R1 Additions (October 2008)

C.11.1 CPW values for the IBM Power Systems - IBM i operating system - model 570 features 7387 and 7388

Table C.11.1. *CPW values for Power System Models*

				Processor CPW				
Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	2 cores	4 cores	8 cores	12 cores	16 cores
570 (9117-MMA)	7387	4.4	2x4MB / 32MB	9850	19400	36200	51500	70000
570 (9117-MMA)	7388	5.0	2x4MB / 32MB	11000	21600	40300	56800	77600

*Note:
1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. Memory speed differences account for some slight variations in performance difference between models.
3. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.11.2 CPW values for the IBM Power Systems - IBM i operating system - model 570 feature 7540

Table C.11.2. *CPW values for Power System Models*

				Processor CPW				
Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	4 cores	8 cores	16 cores	24 cores	32 cores
570 (9117-MMA)	7540	4.2	2x4MB / 32MB	16200	31900	56400	81600	104800

*Note:
1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. Memory speed differences account for some slight variations in performance difference between models.
3. For large partitions, some workloads may experience nonlinear scaling at high system utilization on these new models.
4. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.11.3 CPW values for IBM Power Systems - IBM i operating system - model 560

Table C.11.3. *CPW values for Power System Models*

	Processor CPW

Table C.11.3. CPW values for Power System Models						
Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	4 cores	8 cores	16 cores
560 (8234-EMA)	7537	3.6	2x4MB / 32MB	14100	27600	48500

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. Memory speed differences account for some slight variations in performance difference between models.
3. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.11.4 CPW values for IBM Power Systems - IBM i operating system - models 520 and 550

Table C.11.4. CPW values for Power System Models					
Model	Processor Feature	Chip Speed GHz	L2/L3 cache (1) per chip	CPU (2) Range	Processor CPW
520 (8203-E4A)	5633	4.2	2x4MB / 0MB	1	4300
520 (8203-E4A)	5634	4.2	2x4MB / 0MB	2	8300
520 (8203-E4A)	5635	4.2	2x4MB / 0MB	4	15600
550 (8204-E8A)	4965	3.5	2x4MB / 32MB	2 - 8	7750-27600
550 (8204-E8A)	4966	4.2	2x4MB / 32MB	2 - 8	9200-32650

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. The range of the number of processor cores per system.
3. Memory speed differences account for some slight variations in performance difference between models.
4. CPW values for Power System models introduced in October 2008 were based on IBM i 6.1 plus enhancements in post-release PTFs.

C.12 V6R1 Additions (August 2008)

C.12.1 CPW values for the IBM Power 595 - IBM i operating system using POWER6 processor technology

Table C.12.1. CPW values for Power System Models								
				Processor CPW				
Model	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	8 cores	16 cores	24 cores	32 cores	64 cores (2) (2x32)
595 (9119-FHA)	4695	5000	2x4MB / 32MB	41000	77000	108100	147900	294700
595 (9119-FHA)	4694	4200	2x4MB / 32MB	35500	66400	93800	128000	256200

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache

between two processor cores.
2. This configuration was measured with two 32-core partitions running simultaneously on a 64 core system

C.13 V6R1 Additions (April 2008)

C.13.1 CPW values for IBM Power Systems - IBM i operating system using POWER6 processor technology

Table C.13.1. *CPW values for Power System Models*

Model	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	CPU (2) Range	Processor CPW
520 (9407-M15)	5633	4200	2x4MB / 0MB	1	4300
520 (9408-M25)	5634	4200	2x4MB / 0MB	1 - 2	4300-8300
550 (9409-M50)	4966	4200	2x4MB / 32MB	1 - 4	4800-18000

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. The range of the number of processor cores per system.

C.13.2 CPW values for IBM BladeCenter JS12 - IBM i operating system

Table C.13.2. *IBM BladeCenter models*

Blade Model	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	CPUs(2)	Processor CPW (3)
JS12 (7998-60X)	52BF	3800	2x4MB / 0 MB	1.8 of 2	7100

*Note: 1. These models have a dedicated L2 cache per processor core, and no L3 cache
2. CPW value is for a 1.8-core partition with shared processors and a 0.2-core VIOS partition
3. The value listed is unconstrained CPW (there is sufficient I/O such that the processor would be the first constrained resource). The I/O constrained CPW value for a 12-disk configuration is approximately 1200 CPW (100 CPW per disk).

C.13.3 CPW values for IBM Power Systems - IBM i operating system

Table C.13.3. *CPW values for Power System Models*

				Processor CPW			
Model	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	2 cores	4 cores	8 cores	16 cores
570 (9117-MMA)	5620	3500	2x4MB / 32MB	8150	16100	30100	57600
570 (9117-MMA)	5621/5622	4200	2x4MB / 32MB	9650	19200	35500	68600
570 (9117-MMA)	7380	4700	2x4MB / 32MB	10800	21200	40100	76900

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.

C.14 V6R1 Additions (January 2008)

C.14.1 IBM i5/OS running on IBM BladeCenter JS22 using POWER6 processor technology

Table C.14.1. *IBM BladeCenter models*

Blade Model	Server Feature	Edition Feature	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	CPUs	Processor CPW
JS22 (7998-61X)	n/a	n/a	52BE	4000	2x4MB / 0 MB	3 of 4 (2)	11040
JS22 (7998-61X)	n/a	n/a	52BE	4000	2x4MB / 0 MB	3.7 of 4 (3)	13800

*Note: 1. These models have a dedicated L2 cache per processor core, and no L3 cache
2. CPW value is for a 3-core dedicated partition and a 1-core VIOS
3. CPW value is for a 3.7-core partition with shared processors and a 0.3-core VIOS partition

C.15 V5R4 Additions (July 2007)

C.15.1 IBM System i using the POWER6 processor technology

Table C.15.1. *System i models*

Model	Server Feature	Edition Feature[2]	Processor Feature	Chip Speed MHz	L2/L3 cache (1) per chip	CPU (5) Range	Processor CPW
i570 (9406-MMA)	4910	5460	7380	4700	2x4MB / 32MB	1 - 4	5500-21200
i570 (9406-MMA)	4911	5461	7380	4700	2x4MB / 32MB	2 - 8	10800-40100
i570 (9406-MMA)	4912	5462	7380	4700	2x4MB / 32MB	4 - 16	20100-76900
i570 (9406-MMA)	4922	7053(3)	7380	4700	2x4MB / 32MB	1 - 4	5500-21200
i570 (9406-MMA)	4923	7058(3)	7380	4700	2x4MB / 32MB	1 - 8	5500-40100
i570 (9406-MMA)	4924	7063(3)	7380	4700	2x4MB / 32MB	2 - 16	10800-76900

*Note: 1. These models have a dedicated L2 cache per processor core, and share the L3 cache between two processor cores.
2. This is the Edition Feature for the model. This is the feature displayed when you display the system value QPRCFEAT.
3. Capacity Backup model.
4. Projected values. See Chapter 16 for more information.
5. The range of the number of processor cores per system.

C.16 V5R4 Additions (January/May/August 2006 and January/April 2007)

C.16.1 IBM System i using the POWER5 processor technology

Table C.16.1.1. *System* i models							
Model	Edition Feature[2]	Accelerator Feature	Chip Speed MHz	L2/L3 cache per CPU [(1)]	CPU Range	Processor CPW	5250 OLTP CPW
9406-595	5892	NA	2300	1.9/36MB	32 - 64 [(8)]	108000-216000	Per Processor
9406-595	5872	NA	2300	1.9/36MB	32 - 64 [(8)]	108000-216000	0
9406-595	5891	NA	2300	1.9/36MB	16 - 32	61000-108000	Per Processor
9406-595	5871	NA	2300	1.9/36MB	16 - 32	61000-108000	0
9406-595	5896[(4)]	NA	2300	1.9/36MB	4 - 32	16000-108000	Per Processor
9406-595	5876[(4)]	NA	2300	1.9/36MB	4 - 32	16000-108000	0
9406-595	5890	NA	2300	1.9/36MB	8-16	31500-58800	Per Processor
9406-595	5870	NA	2300	1.9/36MB	8-16	31500-58800	0
9406-595	5895[(4)]	NA	2300	1.9/36MB	2-16	8200-58800	Per Processor
9406-595	5875[(4)]	NA	2300	1.9/36MB	2-16	8200-58800	0
9406-595	7583[(5)]	NA	1900	1.9/36MB	32 - 64 [(8)]	92000-184000	Per Processor
9406-595	7487	NA	1900	1.9/36MB	32 - 64 [(8)]	92000-184000	Per Processor
9406-595	7486	NA	1900	1.9/36MB	32 - 64 [(8)]	92000-184000	0
9406-595	7581[(5)]	NA	1900	1.9/36MB	16 - 32	51000-92000	Per Processor
9406-595	7483	NA	1900	1.9/36MB	16 - 32	51000-92000	Per Processor
9406-595	7482	NA	1900	1.9/36MB	16 - 32	51000-92000	0
9406-595	7590[(4)]	NA	1900	1.9/36MB	4 - 32	13600-92000	Per Processor
9406-595	7912[(4)]	NA	1900	1.9/36MB	4 - 32	13600-92000	Per Processor
9406-595	7580[(5)]	NA	1900	1.9/36MB	8 - 16	26700-50500	Per Processor
9406-595	7481	NA	1900	1.9/36MB	8 - 16	26700-50500	Per Processor
9406-595	7480	NA	1900	1.9/36MB	8 - 16	26700-50500	0
9406-595	7910[(4)]	NA	1900	1.9/36MB	2 - 16	6675-50500	Per Processor
9406-595	7911[(4)]	NA	1900	1.9/36MB	2 - 16	6675-50500	Per Processor
9406-570	7760[(4)]	NA	2200	1.9/36MB	2 - 16	8100-58500	Per Processor
9406-570	7918[(4)]	NA	2200	1.9/36MB	2 - 16	8100-58500	Per Processor
9406-570	7765[(5)]	NA	2200	1.9/36MB	8 - 16	31100-58500	Per Processor
9406-570	7749	NA	2200	1.9/36MB	8 - 16	31100-58500	Per Processor
9406-570	7759	NA	2200	1.9/36MB	8 - 16	31100-58500	0
9406-570	7764[(5)]	NA	2200	1.9/36MB	4 - 8	16700-31100	Per Processor
9406-570	7748	NA	2200	1.9/36MB	4 - 8	16700-31100	Per Processor
9406-570	7758	NA	2200	1.9/36MB	4 - 8	16700-31100	0
9406-570	7916[(4)]	NA	2200	1.9/36MB	1 - 8	4200-31100	Per Processor
9406-570	7917[(4)]	NA	2200	1.9/36MB	1 - 8	4200-31100	Per Processor
9406-570	7763[(5)]	NA	2200	1.9/36MB	2 - 4	8400-16000	Per Processor
9406-570	7747	NA	2200	1.9/36MB	2 - 4	8400-16000	Per Processor
9406-570	7757	NA	2200	1.9/36MB	2 - 4	8400-16000	0
9406-570	7914[(4)]	NA	2200	1.9/36MB	1 - 4	4200-16000	Per Processor
9406-570	7915[(4)]	NA	2200	1.9/36MB	1 - 4	4200-16000	Per Processor
9406-550	7551[(5)]	NA	1900	1.9/36MB	1 - 4	3800-14000	Per Processor
9406-550	7629[(6)]	NA	1900	1.9/36MB	1 - 4	3800-14000	0
9406-550	7155	NA	1900	1.9/36MB	1 - 4	3800-14000	Per Processor
9406-550	7154	NA	1900	1.9/36MB	1 - 4	3800-14000	0
9406-550	7920[(4)]	NA	1900	1.9/36MB	1 - 4	3800-14000	Per Processor
9406-550	7921[(4)]	NA	1900	1.9/36MB	1 - 4	3800-14000	Per Processor
9406-525	7792[(11)]	NA	1900	1.9/36MB	1-2	3800-7100	3800-7100
9406-525	7791[(11)]	NA	1900	1.9/36MB	1-2	3800-7100	3800-7100
9406-525	7790[(11)]	NA	1900	1.9/36MB	1-2	3800-7100	3800-7100
9407-515	6028[(11)]	NA	1900	1.9/36MB	2	7100[(12)]	7100
9407-515	6021[(11)]	NA	1900	1.9/36MB	2	7100[(12)]	7100
9407-515	6018[(11)]	NA	1900	1.9/36MB	1	3800[(12)]	3800
9407-515	6011[(11)]	NA	1900	1.9/36MB	1	3800[(12)]	3800

Table C.16.1.1. System i models							
Model	Edition Feature[2]	Accelerator Feature	Chip Speed MHz	L2/L3 cache per CPU [(1)]	CPU Range	Processor CPW	5250 OLTP CPW
9407-515	6010[(11)]	NA	1900	1.9/36MB	1	3800[(12)]	3800
9406-520	7375[(5)]	NA	1900	1.9/36MB	1 - 2	3800-7100	3800-7100
9406-520	7736	NA	1900	1.9/36MB	1 - 2	3800-7100	3800-7100
9406-520	7785	NA	1900	1.9/36MB	1 - 2	3800-7100	0
9406-520	7784	NA	1900	1.9/36MB	1	3800	0
9406-520	7691[(10)]	NA	1900	1.9/36MB	1	3800	0
9406-520	7374[(5)]	NA	1900	1.9/36MB	1[(3)]	2800	2800
9406-520	7735	NA	1900	1.9/36MB	1[(3)]	2800	2800
9406-520	7373[(5)]	NA	1900	1.9/36MB	1[(3)]	1200	1200
9406-520	7734	NA	1900	1.9/36MB	1[(3)]	1200	1200
Value							
9406-520	7352	7357	1900	1.9/36MB	1[(3)]	1200-3800 [9]	60
9406-520	7350	7355	1900	1.9MB/NA	1[(3)]	600-3100 [9]	30
Express							
9405-520	7152	NA	1900	1.9/36MB	1	3800	60
9405-520	7144	NA	1900	1.9/36MB	1	3800	60
9405-520	7143	7354	1900	1.9/36MB	1[(3)]	1200-3800 [9]	60
9405-520	7148	7687	1900	1.9/36MB	1[(3)]	1200-3800 [9]	60
9405-520	7156	7353	1900	1.9/NA	1[(3)]	600-3100 [9]	30
9405-520	7142	7682	1900	1.9MB/NA	1[(3)]	600-3100 [9]	30
9405-520	7141	7681	1900	1.9MB/NA	1[(3)]	600-3100 [9]	30
9405-520	7140	7680	1900	1.9MB/NA	1[(3)]	600-3100 [9]	30

*Note:
1. These models share L2 and L3 cache between two processor cores.
2. This is the Edition Feature for the model. This is the feature displayed when you display the system value QPRCFEAT.
3. CPU Range - entry model is a partial processor model, offering multiple price/performance points for the entry market.
4. Capacity Backup model.
5. High Availability model.
6. Domino edition.
7. NR - Not Recommended: the 600 CPW processor offering is not recommended for Domino.
8. The 64-way CPW value is reflects two 32-way partitions.
9. These models are accelerator models. The base CPW value is the capacity with the default processor feature. The max CPW value is the capacity when purchasing the accelerator processor feature.
10. Collaboration Edition. (Announced May 9, 2006)
11. User based pricing models.
12. These values listed are unconstrained CPW values (there is sufficient I/O such that the processor would be the first constrained resource). The I/O constrained CPW value for an 8-disk configuration is approximately 800 CPW (100 CPW per disk).

C.17 V5R3 Additions (May, July, August, October 2004, July 2005)

New for this release is the eServer i5 servers which provide a significant performance improvement when compared to iSeries model 8xx servers.

C.17.1 IBM @server® i5 Servers

Table C.17.1.1. @server® i5 Servers

Model	Chip Speed MHz	L2 cache per CPU (1)	L3 cache per CPU(2)	CPU Range	Processor CPW	5250 OLTP CPW
595-0952 (7485)	1650	1.9 MB	36 MB	32 - 64 (7)	86000-165000	12000-165000
595-0952 (7484)	1650	1.9 MB	36 MB	32 - 64 (7)	86000-165000	0
595-0947 (7499)	1650	1.9 MB	36 MB	16 - 32	46000-85000	12000-85000
595-0947 (7498)	1650	1.9 MB	36 MB	16 - 32	46000-85000	0
595-0946 (7497)	1650	1.9 MB	36 MB	8 - 16	24500-45500	12000-45500
595-0946 (7496)	1650	1.9 MB	36 MB	8 - 16	24500-45500	0
570-0926 (7476)	1650	1.9 MB	36 MB	13 - 16	36300-44700	12,000-44,700
570-0926 (7475)	1650	1.9 MB	36 MB	13 - 16	36300-44700	0
570-0926 (7563)[5]	1650	1.9 MB	36 MB	13 - 16	36300-44700	12000-44,700
570-0928 (7570)[4]	1650	1.9 MB	36 MB	2 - 16	6350-44700	6,350-44,700
570-0928 (7474)	1650	1.9 MB	36 MB	9 - 12	25500-33400	12,000-33,400
570-0924 (7473)	1650	1.9 MB	36 MB	9 - 12	25500-33400	0
570-0924 (7562)[5]	1650	1.9 MB	36 MB	9 - 12	25500-33400	12000-44,700
570-0922 (7472)	1650	1.9 MB	36 MB	5 - 8	15200-23500	12,000-23,500
570-0922 (7471)	1650	1.9 MB	36 MB	5 - 8	15200-23500	0
570-0922 (7561)[5]	1650	1.9 MB	36 MB	5 - 8	15200-23500	12,000-23,500
570-0921 (7495)	1650	1.9 MB	36 MB	2 - 4	6350-12000	12000
570-0921 (7494)	1650	1.9 MB	36 MB	2 - 4	6350-12000	0
570-0921 (7560)[5]	1650	1.9 MB	36 MB	2 - 4	6350-12000	12000
570-0930 (7491)	1650	1.9 MB	36 MB	1 - 2	3300-6000	6000
570-0930 (7490)	1650	1.9 MB	36 MB	1 - 2	3300-6000	0
570-0930 (7559)[5]	1650	1.9 MB	36 MB	1 - 2	3300-6000	6,000
570-0920 (7470)	1650	1.9 MB	36 MB	2 - 4	6350-12000	Max
570-0920 (7469)	1650	1.9 MB	36 MB	2 - 4	6350-12000	0
570-0919 (7489)	1650	1.9 MB	36 MB	1 - 2	3300-6000	Max
570-0919 (7488)	1650	1.9 MB	36 MB	1 - 2	3300-6000	0
550-0915 (7530)[6]	1650	1.9 MB	36 MB	2 - 4	6350-12000	0
550-0915 (7463)	1650	1.9 MB	36 MB	1 - 4	3300-12000	3,300-12,000
550-0915 (7462)	1650	1.9 MB	36 MB	1 - 4	3300-12000	0
550-0915 (7558)	1650	1.9 MB	36 MB	1 - 4	3300-12000	3,300-12,000
520-0905 (7457)	1650	1.9 MB	36 MB	2	6000	3,300-6000
520-0905 (7456)	1650	1.9 MB	36 MB	2	6000	0
520-0905 (7555)[5]	1650	1.9 MB	36 MB	2	6000	3,300-6,000
520-0904 (7455)	1650	1.9 MB	36 MB	1	3300	3,300
520-0904 (7454)	1650	1.9 MB	36 MB	1	3300	0
520-0904 (7554)[5]	1650	1.9 MB	36 MB	1	3300	3,300
520-0903 (7453)	1500	1.9 MB	NA	1	2400	2400
520-0912 (7397)	1500	1.9 MB	NA	1	2400	60
520-0912 (7395)	1500	1.9 MB	NA	1	2400	60
520-0903 (7452)	1500	1.9 MB	NA	1	2400	0
520-0903 (7553)[5]	1500	1.9 MB	NA	1	2400	2400
520-0902 (7459)	1500	1.9 MB	NA	1 (3)	1000	1000
520-0902 (7458)	1500	1.9 MB	NA	1 (3)	1000	0
520-0902 (7552)[5]	1500	1.9 MB	NA	1 (3)	1000	1000
520-0901 (7451)	1500	1.9MB	NA	1 (3)	1000	60
520-0900 (7450)	1500	1.9 MB	NA	1 (3)	500	30

*Note: 1. 1.9MB - These models share L2 cache between 2 processors.
2. 36MB - These models share L3 cache between 2 processors.
3. CPU Range - Partial processor models, offering multiple price/performance points for the entry market.

4. Capacity Backup model.
5. High Availability model.
6. Domino edition.
7. The 64-way is measured as two 32-way partitions since i5/OS does not support a 64-way partition.

C.18 V5R2 Additions (February, May, July 2003)

New for this release is a product line refresh of the iSeries hardware which simplifies the model structure and minimizes the number of interactive choices. In most cases, the customer must choose between a Standard edition which includes a 5250 interactive CPW value of 0, or an Enterprise edition which supports the maximum 5250 OLTP capacity. The table in the following section lists the entire product line for 2003.

C.18.1 iSeries Model 8xx Servers

Table C.18.1.1. iSeries Models 8xx Servers

Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	5250 OLTP CPW*
890-2498 (7427)	1300	1.41 MB*	24 - 32	29300-37400	Max
890-2498 (7425)	1300	1.41 MB*	24 - 32	29300-37400	0
890-2497 (7424)	1300	1.41 MB*	16 - 24	20000-29300	Max
890-2497 (7422)	1300	1.41 MB*	16 - 24	20000-29300	0
870-2486 (7421)	1300	1.41 MB*	8 - 16	11500-20000	Max
870-2486 (7419)	1300	1.41 MB*	8 - 16	11500-20000	0
870-2489 (7431)	1300	1.41 MB*	5 - 8	7700-11500	0
870-2489 (7433)	1300	1.41 MB*	5 - 8	7700-11500	Max
825-2473 (7418)	1100	1.41 MB*	3 - 6	3600-6600	Max
825-2473 (7416)	1100	1.41 MB*	3 - 6	3600-6600	0
810-2469 (7430)	750	4 MB	2	2700	Max
810-2469 (7428)	750	4 MB	2	2700	0
810-2467 (7412)	750	4 MB	1	1470	Max
810-2467 (7410)	750	4 MB	1	1470	0
810-2466 (7409)	540	2 MB	1	1020	Max
810-2466 (7407)	540	2 MB	1	1020	0
810-2465 (7406)	540	2 MB	1	750	Max
810-2465 (7404)	540	2 MB	1	750	0
800-2464 (7408)	540	2 MB	1	950	50
800-2463 (7400)	540	0 MB	1	300	25

*Note: 1. 5250 OLTP CPW - Max (maximum CPW value). There is no limit on 5250 OLTP workloads and the full capacity of the server (Processor CPW) is available for 5250 OLTP work.
2. 1.41MB - These models share L2 cache between 2 processors

C.18.2 Model 810 and 825 iSeries for Domino (February 2003)

Table C.18.2.1. iSeries for Domino 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	5250 OLTP CPW*
825-2473 (7416)	1100	1.41 MB	6	6600	0
825-2473 (7416)	1100	1.41 MB	4	na	0
810-2469 (7428)	750	4 MB	2	2700	0
810-2467 (7410)	750	4 MB	1	1470	0
810-2466 (7407)	540	2 MB	1	1020	0

*Note: 1. 5250 OLTP CPW - With a rating of 0, adequate interactive processing is available for a single 5250 job to perform system administration functions.
2. na - indicates the rating is not available for the 4-way processor configuration

C.19 V5R2 Additions

In V5R2 the following new iSeries models were introduced:
- 890 Base and Standard models
- 840 Base models
- 830 Base and Standard models

Base models represent server systems with "0" interactive capability. Standard Models represent systems that have interactive features available and also may have Capacity Upgrade on Demand Capability.

C.19.1 Base Models 8xx Servers

Table C.19.1.1 Base Models 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
890-0198 (none)	1300	1.41 MB*	32	37400	0
890-0197 (none)	1300	1.41 MB*	24	29300	0
840-0159 (none)	600	16 MB	24	20200	0
840-0158 (none)	600	16 MB	12	12000	0
830-0153 (none)	540	4 MB	8	7350	0

* 890 Models share L2 cache between 2 processors

C.19.2 Standard Models 8xx Servers

Standard models have an initial offering of processor and interactive capacity with featured upgrades for activation of additional processors and increased interactive capacity. Processor features are offered through Capacity Upgrade on Demand, described in ***C.20 V5R1 Additions***.

Table C.19.2.1 Standard Models 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW
890-2488 (1576)	1300	1.41 MB*	24 - 32	29300-37400	120
890-2488 (1577)	1300	1.41 MB*	24 - 32	29300-37400	240
890-2488 (1578)	1300	1.41 MB*	24 - 32	29300-37400	560

Table C.19.2.1 Standard Models 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW
890-2488 (1579)	1300	1.41 MB*	24 - 32	29300-37400	1050
890-2488 (1581)	1300	1.41 MB*	24 - 32	29300-37400	2000
890-2488 (1583)	1300	1.41 MB*	24 - 32	29300-37400	4550
890-2488 (1585)	1300	1.41 MB*	24 - 32	29300-37400	10000
890-2488 (1587)	1300	1.41 MB*	24 - 32	29300-37400	16500
890-2488 (1588)	1300	1.41 MB*	24 - 32	29300-37400	20200
890-2488 (1591)	1300	1.41 MB*	24 - 32	29300-37400	37400
890-2487 (1576)	1300	1.41 MB*	16 - 24	20000-29300	120
890-2487 (1577)	1300	1.41 MB*	16 - 24	20000-29300	240
890-2487 (1578)	1300	1.41 MB*	16 - 24	20000-29300	560
890-2487 (1579)	1300	1.41 MB*	16 - 24	20000-29300	1050
890-2487 (1581)	1300	1.41 MB*	16 - 24	20000-29300	2000
890-2487 (1583)	1300	1.41 MB*	16 - 24	20000-29300	4550
890-2487 (1585)	1300	1.41 MB*	16 - 24	20000-29300	10000
890-2487 (1587)	1300	1.41 MB*	16 - 24	20000-29300	16500
890-2487 (1588)	1300	1.41 MB*	16 - 24	20000-29300	20200
830-2349 (1531)	540	4 MB	4 - 8	4200-7350	70
830-2349 (1532)	540	4 MB	4 - 8	4200-7350	120
830-2349 (1533	540	4 MB	4 - 8	4200-7350	240
830-2349 (1534)	540	4 MB	4 - 8	4200-7350	560
830-2349 (1535)	540	4 MB	4 - 8	4200-7350	1050
830-2349 (1536)	540	4 MB	4 - 8	4200-7350	2000
830-2349 (1537)	540	4 MB	4 - 8	4200-7350	4550

* 890 Models share L2 cache between 2 processors

Other models available in V5R2 and listed in ***C.20 V5R1 Additions*** are as follows:

- All 270 Models
- All 820 Models
- Model 830-2400
- All 840 model listed in ***Table C.20.4.1.1 V5R1 Capacity Upgrade on-demand Models***

C.20 V5R1 Additions

In V5R1 the following new iSeries models were introduced:

- 820 and 840 server models
- 270 server models
- 270 and 820 Dedicated servers for Domino
- 840 Capacity Upgrade on-demand models (including V4R5 models December 2000)

.

C.20.1 Model 8xx Servers

Table C.19.1.1 Model 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
820-0150 (none)	600	2 MB	1	1100	0
820-0151 (none)	600	4 MB	2	2350	0
820-0152 (none)	600	4 MB	4	3700	0
820-2435 (1521)	600	2 MB	1	600	35
820-2435 (1522)	600	2 MB	1	600	70
820-2435 (1523)	600	2 MB	1	600	120
820-2435 (1524)	600	2 MB	1	600	240
820-2436 (1521)	600	2 MB	1	1100	35
820-2436 (1522)	600	2 MB	1	1100	70
820-2436 (1523)	600	2 MB	1	1100	120
820-2436 (1524)	600	2 MB	1	1100	240
820-2436 (1525)	600	2 MB	1	1100	560
820-2437 (1521)	600	4 MB	2	2350	35
820-2437 (1522)	600	4 MB	2	2350	70
820-2437 (1523)	600	4 MB	2	2350	120
820-2437 (1524)	600	4 MB	2	2350	240
820-2437 (1525)	600	4 MB	2	2350	560
820-2437 (1526)	600	4 MB	2	2350	1050
820-2438 (1521)	600	4 MB	4	3700	35
820-2438 (1522)	600	4 MB	4	3700	70
820-2438 (1523)	600	4 MB	4	3700	120
820-2438 (1524)	600	4 MB	4	3700	240
820-2438 (1525)	600	4 MB	4	3700	560
820-2438 (1526)	600	4 MB	4	3700	1050
820-2438 (1527)	600	4 MB	4	3700	2000
830-2400 (1531)	400	2 MB	2	1850	70
830-2400 (1532)	400	2 MB	2	1850	120
830-2400 (1533)	400	2 MB	2	1850	240
830-2400 (1534)	400	2 MB	2	1850	560
830-2400 (1535)	400	2 MB	2	1850	1050
830-2402 (1531)	540	4 MB	4	4200	70
830-2402 (1532)	540	4 MB	4	4200	120
830-2402 (1533)	540	4 MB	4	4200	240
830-2402 (1534)	540	4 MB	4	4200	560
830-2402 (1535)	540	4 MB	4	4200	1050
830-2402 (1536)	540	4 MB	4	4200	2000
830-2403 (1531)	540	4 MB	8	7350	70
830-2403 (1532)	540	4 MB	8	7350	120
830-2403 (1533)	540	4 MB	8	7350	240
830-2403 (1534)	540	4 MB	8	7350	560
830-2403 (1535)	540	4 MB	8	7350	1050
830-2403 (1536)	540	4 MB	8	7350	2000
830-2403 (1537)	540	4 MB	8	7350	4550
840-2461 (1540)	600	16 MB	24	20200	120
840-2461 (1541)	600	16 MB	24	20200	240
840-2461 (1542)	600	16 MB	24	20200	560
840-2461 (1543)	600	16 MB	24	20200	1050
840-2461 (1544)	600	16 MB	24	20200	2000

Table C.19.1.1 Model 8xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
840-2461 (1545)	600	16 MB	24	20200	4550
840-2461 (1546)	600	16 MB	24	20200	10000
840-2461 (1547)	600	16 MB	24	20200	16500
840-2461 (1548)	600	16 MB	24	20200	20200

Note: 830 models were first available in V4R5.

C.20.2 Model 2xx Servers

Table C.20.2.1 Model 2xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
270-2431 (1518)	540	n/a	1	465	30
270-2432 (1516)	540	2 MB	1	1070	0
270-2432 (1519)	540	2 MB	1	1070	50
270-2434 (1516)	600	4 MB	2	2350	0
270-2434 (1520)	600	4 MB	2	2350	70

C.20.3 V5R1 Dedicated Server for Domino

Table C.20.3 .1 Dedicated Servers for Domino					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	NonDomino CPW	Interactive CPW
270-2452 (none)	540	2 MB	1	100	0
270-2454 (none)	600	4 MB	2	240	0
820-2456 (none)	600	2 MB	1	120	0
820-2457 (none)	600	4 MB	2	240	0
820-2458 (none)	600	4 MB	4	380	0

C.20.4 Capacity Upgrade on-demand Models

New in V4R5 (December 2000) , Capacity Upgrade on Demand (CUoD) capability offered for the iSeries Model 840 enables users to start small, then increase processing capacity without disrupting any of their current operations. To accomplish this, six processor features are available for the Model 840. These new processor features offer a Startup number of active processors; 8-way, 12-way or 18-way , with additional On-Demand processors capacity built-in (Standby). The customer can add capacity in increments of one processor (or more), up to the maximum number of On-Demand processors built into the Model 840. CUoD has significant value for installations who want to upgrade without disruption. To activate processors, the customer simply enters a unique activation code ("software key") at the server console (DST/SST screen).

The table below list the Capacity Upgrade on Demand features.

	Startup Processors ("Active")	On-Demand Processors ("Stand-by")	TOTAL Processors
840-2352 (2416)	8	4	12
840-2353 (2417)	12	6	18
840-2354 (2419)	18	6	24

Note: Features 23xx added in V5R1. Features 24xx were available in V4R5 (December 2000)

C.20.4.1 CPW Values and Interactive Features for CUoD Models

The following tables list only the processor CPW value for the Startup number of processors as well as a processor CPW value that represents the full capacity of the server for all processors active (Startup + On-Demand). Interpolation between these values can give an approximate rating for incremental processor improvements, although the incremental improvements will vary by workload and because earlier activations may take advantage of caching resources that are shared among processors.

Interactive Features are available for the Model 840 ordered with CUoD Processor Features. Interactive performance is limited by total capacity of the active processors . When ordering FC 1546, FC 1547, or FC 1548 one should consider that the full capacity of interactive is not available unless all of the On-Demand processors have been activated .For more information on Capacity Upgrade on-demand, see URL: : http://www-1.ibm.com/servers/eserver/iseries/hardware/ondemand

Note: In V5R2, CUoD features come with all standard models, which are described in the **V5R2 Additions** section of this appendix.

Table C.20.4.1.1 V5R1 Capacity Upgrade on-demand Models					
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW
840-2352 (1540)	600	16 MB	8 - 12	9000 - 12000	120
840-2352 (1541)	600	16 MB	8 - 12	9000 - 12000	240
840-2352 (1542)	600	16 MB	8 - 12	9000 - 12000	560
840-2352 (1543)	600	16 MB	8 - 12	9000 - 12000	1050
840-2352 (1544)	600	16 MB	8 - 12	9000 - 12000	2000
840-2352 (1545)	600	16 MB	8 - 12	9000 - 12000	4550
840-2352 (1546)	600	16 MB	8 - 12	9000 - 12000	10000
840-2353 (1540)	600	16 MB	12 - 18	12000 - 16500	120
840-2353 (1541)	600	16 MB	12 - 18	12000 - 16500	240
840-2353 (1542)	600	16 MB	12 - 18	12000 - 16500	560
840-2353 (1543)	600	16 MB	12 - 18	12000 - 16500	1050
840-2353 (1544)	600	16 MB	12 - 18	12000 - 16500	2000
840-2353 (1545)	600	16 MB	12 - 18	12000 - 16500	4550
840-2353 (1546)	600	16 MB	12 - 18	12000 - 16500	10000
840-2353 (1547)	600	16 MB	12 - 18	12000 - 16500	16500
840-2354 (1540)	600	16 MB	18 - 24	16500 - 20200	120
840-2354 (1541)	600	16 MB	18 - 24	16500 - 20200	240
840-2354 (1542)	600	16 MB	18 - 24	16500 - 20200	560
840-2354 (1543)	600	16 MB	18 - 24	16500 - 20200	1050
840-2354 (1544)	600	16 MB	18 - 24	16500 - 20200	2000
840-2354 (1545)	600	16 MB	18 - 24	16500 - 20200	4550
840-2354 (1546)	600	16 MB	18 - 24	16500 - 20200	10000
840-2354 (1547)	600	16 MB	18 - 24	16500 - 20200	16500
840-2354 (1548)	600	16 MB	18 - 24	16500 - 20200	20200

Table C.20.4.1.2 V4R5 Capacity Upgrade on-demand Models (12/00)					
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW
840-2416 (1540)	500	8 MB	8 - 12	7800 - 10000	120
840-2416 (1541)	500	8 MB	8 - 12	7800 - 10000	240
840-2416 (1542)	500	8 MB	8 - 12	7800 - 10000	560
840-2416 (1543)	500	8 MB	8 - 12	7800 - 10000	1050
840-2416 (1544)	500	8 MB	8 - 12	7800 - 10000	2000
840-2416 (1545)	500	8 MB	8 - 12	7800 - 10000	4550
840-2416 (1546)	500	8 MB	8 - 12	7800 - 10000	10000
840-2417 (1540)	500	8 MB	12 - 18	10000 - 13200	120
840-2417 (1541)	500	8 MB	12 - 18	10000 - 13200	240
840-2417 (1542)	500	8 MB	12 - 18	10000 - 13200	560
840-2417 (1543)	500	8 MB	12 - 18	10000 - 13200	1050
840-2417 (1544)	500	8 MB	12 - 18	10000 - 13200	2000
840-2417 (1545)	500	8 MB	12 - 18	10000 - 13200	4550
840-2417 (1546)	500	8 MB	12 - 18	10000 - 13200	10000
840-2419 (1540)	500	8 MB	18 - 24	13200 - 16500	120
840-2419 (1541)	500	8 MB	18 - 24	13200 - 16500	240
840-2419 (1542)	500	8 MB	18 - 24	13200 - 16500	560
840-2419 (1543)	500	8 MB	18 - 24	13200 - 16500	1050
840-2419 (1544)	500	8 MB	18 - 24	13200 - 16500	2000
840-2419 (1545)	500	8 MB	18 - 24	13200 - 16500	4550
840-2419 (1546)	500	8 MB	18 - 24	13200 - 16500	10000
840-2419 (1547)	500	8 MB	18 - 24	13200 - 16500	16500

C.21 V4R5 Additions

For the V4R5 hardware additions, the tables show each new server model characteristics and its maximum interactive CPW capacity. For previously existing hardware, the tables show for each server model the maximum interactive CPW and its corresponding CPU % and the point (the knee of the curve) where the interactive utilization begins to increasingly impact client/server performance. For the models that have multiple processors, and the knee of the curve is also given in CPU%, the percent value is the percent of all the processors (not of a single one).
CPW values may be increased as enhancements are made to the operating system (e.g. each feature of the Model 53S for V3R7 and V4R1). The server model behavior is fixed to the original CPW values.
For example, the model 53S-2157 had V3R7 CPWs of 509.9/30.7 and V4R1 CPWs 650.0/32.2. When using the 53S with V4R1, this means the knee of the curve is 2.6% CPU and the maximum interactive is 7.7% CPU, the same as it was in V3R7.

The 2xx, 8xx and SBx models are new in V4R5.

C.21.1 AS/400e Model 8xx Servers

Table C.21.1 Model 8xx Servers (all new Condor models)

Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW**
820-2395 (1521)	400	n/a	1	**370**	**35**
820-2395 (1522)	400	n/a	1	**370**	**70**
820-2395 (1523)	400	n/a	1	**370**	**120**
820-2395 (1524)	400	n/a	1	**370**	**240**
820-2396 (1521)	450	2 MB	1	**950**	**35**
820-2396 (1522)	450	2 MB	1	**950**	**70**
820-2396 (1523)	450	2 MB	1	**950**	**120**
820-2396 (1524)	450	2 MB	1	**950**	**240**
820-2396 (1525)	450	2 MB	1	**950**	**560**
820-2397 (1521)	500	4 MB	2	**2000**	**35**
820-2397 (1522)	500	4 MB	2	**2000**	**70**
820-2397 (1523)	500	4 MB	2	**2000**	**120**
820-2397 (1524)	500	4 MB	2	**2000**	**240**
820-2397 (1525)	500	4 MB	2	**2000**	**560**
820-2397 (1526)	500	4 MB	2	**2000**	**1050**
820-2398 (1521)	500	4 MB	4	**3200**	**35**
820-2398 (1522)	500	4 MB	4	**3200**	**70**
820-2398 (1523)	500	4 MB	4	**3200**	**120**
820-2398 (1524)	500	4 MB	4	**3200**	**240**
820-2398 (1525)	500	4 MB	4	**3200**	**560**
820-2398 (1526)	500	4 MB	4	**3200**	**1050**
820-2398 (1527)	500	4 MB	4	**3200**	**2000**
830-2400 (1531)	400	2 MB	2	**1850**	**70**
830-2400 (1532)	400	2 MB	2	**1850**	**120**
830-2400 (1533)	400	2 MB	2	**1850**	**240**
830-2400 (1534)	400	2 MB	2	**1850**	**560**
830-2400 (1535)	400	2 MB	2	**1850**	**1050**
830-2402 (1531)	540	4 MB	4	**4200**	**70**
830-2402 (1532)	540	4 MB	4	**4200**	**120**
830-2402 (1533)	540	4 MB	4	**4200**	**240**
830-2402 (1534)	540	4 MB	4	**4200**	**560**

Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW**
830-2402 (1535)	540	4 MB	4	**4200**	**1050**
830-2402 (1536)	540	4 MB	4	**4200**	**2000**
830-2403 (1531)	540	4 MB	8	**7350**	**70**
830-2403 (1532)	540	4 MB	8	**7350**	**120**
830-2403 (1533)	540	4 MB	8	**7350**	**240**
830-2403 (1534)	540	4 MB	8	**7350**	**560**
830-2403 (1535)	540	4 MB	8	**7350**	**1050**
830-2403 (1536)	540	4 MB	8	**7350**	**2000**
830-2403 (1537)	540	4 MB	8	**7350**	**4550**
840-2418 (1540)	500	8 MB	12	**10000**	**120**
840-2418 (1541)	500	8 MB	12	**10000**	**240**
840-2418 (1542)	500	8 MB	12	**10000**	**560**
840-2418 (1543)	500	8 MB	12	**10000**	**1050**
840-2418 (1544)	500	8 MB	12	**10000**	**2000**
840-2418 (1545)	500	8 MB	12	**10000**	**4550**
840-2418 (1546)	500	8 MB	12	**10000**	**10000**
840-2420 (1540)	500	8 MB	24	**16500**	**120**
840-2420 (1541)	500	8 MB	24	**16500**	**240**
840-2420 (1542)	500	8 MB	24	**16500**	**560**
840-2420 (1543)	500	8 MB	24	**16500**	**1050**
840-2420 (1544)	500	8 MB	24	**16500**	**2000**
840-2420 (1545)	500	8 MB	24	**16500**	**4550**
840-2420 (1546)	500	8 MB	24	**16500**	**10000**
840-2420 (1547)	500	8 MB	24	**16500**	**16500**

C.21.2 Model 2xx Servers

Table C.21.2.1 Model 2xx Servers

Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW**
250-2295	200	n/a	1	**50**	**15**
250-2296	200	n/a	1	**75**	**20**
270-2248 (1517)	400	n/a	1	**150**	**25**
270-2250 (1516)	400	n/a	1	**370**	**0**
270-2250 (1518)	400	n/a	1	**370**	**30**
270-2252 (1516)	450	2 MB	1	**950**	**0**
270-2252 (1519)	450	2 MB	1	**950**	**50**
270-2253 (1516)	450	4 MB	2	**2000**	**0**
270-2253 (1520)	450	4 MB	2	**2000**	**70**

C.21.3 Dedicated Server for Domino

Table C.21.3.1 Dedicated Server for Domino

Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Non Domino CPW**	**Interactive CPW**
820-2425	450	2 MB	1	**100**	**0**
820-2426	500	4 MB	2	**200**	**0**
820-2427	500	4 MB	4	**300**	**0**
270-2422	400	n/a	1	**50**	**0**
270-2423	450	2 MB	1	**100**	**0**
270-2424	450	4 MB	2	**200**	**0**

C.21.4 SB Models

Table C.21.4.1 SB Models					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW***	**Interactive CPW**
SB2-2315	540	4 MB	8	**7350**	**70**
SB3-2316	500	8 MB	12	**10000**	**120**
SB3-2318	500	8 MB	24	**16500**	**120**

* Note: The "Processor CPW" values listed for the SB models are identical to the 830-2403-1531 (8-way), the 840-2418-1540 (12-way) and the 840-2420-1540 (24-way). However, due to the limited disk and memory of the SB models, it would not be possible to measure these values using the CPW workload. Disk space is not a high priority for middle-tier servers performing CPU-intensive work because they are always connected to another computer acting as the "database" server in a multi-tier implementation.

C.22 V4R4 Additions

The Model 7xx is new in V4R4. Also in V4R4 are the Model 170s features 2289 and 2388 were added. Testing in the Rochester laboratory has shown that for systems executing traditional commercial applications such as RPG or COBOL interactive general business applications may experience about a 5% increase in CPU requirements. This effect was observed using the workload used to compute CPW, as shown in the tables that follows. Except for systems which are nearing the need for an upgrade, we do not expect this increase to significantly affect transaction response times. It is recommended that other sections of the Performance Capabilities Reference Manual (or other sizing and positioning documents) be used to estimate the impact of upgrading to the new release.

C.22.1 AS/400e Model 7xx Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

Table C.22.1.1 Model 7xx Servers (all new Northstar models)						
Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW (Knee)**	Interactive CPW (Max)
720-2061 (Base)	200	n/a	1	**240**	**35**	40.8
720-2061 (1501)	200	n/a	1	**240**	**70**	81.7
720-2061 (1502)	200	n/a	1	**240**	**120**	140
720-2062 (Base)	200	4 MB	1	**420**	**35**	40.8
720-2062 (1501)	200	4 MB	1	**420**	**70**	81.7
720-2062 (1502)	200	4 MB	1	**420**	**120**	140
720-2062 (1503)	200	4 MB	1	**420**	**240**	280
720-2063 (Base)	200	4 MB	2	**810**	**35**	40.8
720-2063 (1502)	200	4 MB	2	**810**	**120**	140
720-2063 (1503)	200	4 MB	2	**810**	**240**	280
720-2063 (1504)	200	4 MB	2	**810**	**560**	653.3
720-2064 (Base)	255	4 MB	4	**1600**	**35**	40.8
720-2064 (1502)	255	4 MB	4	**1600**	**120**	140
720-2064 (1503)	255	4 MB	4	**1600**	**240**	280
720-2064 (1504)	255	4 MB	4	**1600**	**560**	653.3
720-2064 (1505)	255	4 MB	4	**1600**	**1050**	1225
730-2065 (Base)	262	4 MB	1	**560**	**70**	81.7
730-2065 (1507)	262	4 MB	1	**560**	**120**	140

Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW (Knee)**	Interactive CPW (Max)
730-2065 (1508)	262	4 MB	1	**560**	**240**	280
730-2065 (1509)	262	4 MB	1	**560**	**560**	653.3
730-2066 (Base)	262	4 MB	2	**1050**	**70**	81.7
730-2066 (1507)	262	4 MB	2	**1050**	**120**	140
730-2066 (1508)	262	4 MB	2	**1050**	**240**	280
730-2066 (1509)	262	4 MB	2	**1050**	**560**	653.3
730-2066 (1510)	262	4 MB	2	**1050**	**1050**	1225
730-2067 (Base)	262	4 MB	4	**2000**	**70**	81.7
730-2067 (1508)	262	4 MB	4	**2000**	**240**	280
730-2067 (1509)	262	4 MB	4	**2000**	**560**	653.3
730-2067 (1510)	262	4 MB	4	**2000**	**1050**	1225
730-2067 (1511)	262	4 MB	4	**2000**	**2000**	2333.3
730-2068 (Base)	262	4 MB	8	**2890**	**70**	81.7
730-2068 (1508)	262	4 MB	8	**2890**	**240**	280
730-2068 (1509)	262	4 MB	8	**2890**	**560**	653.3
730-2068 (1510)	262	4 MB	8	**2890**	**1050**	1225
730-2068 (1511)	262	4 MB	8	**2890**	**2000**	2333.3
740-2069 (Base)	262	8 MB	8	**3660**	**120**	140
740-2069 (1510)	262	8 MB	8	**3660**	**1050**	1225
740-2069 (1511)	262	8 MB	8	**3660**	**2000**	2333.3
740-2069 (1512)	262	8 MB	8	**3660**	**3660**	4270
740-2070 (Base)	262	8 MB	12	**4550**	**120**	140
740-2070 (1510)	262	8 MB	12	**4550**	**1050**	1225
740-2070 (1511)	262	8 MB	12	**4550**	**2000**	2333.3
740-2070 (1512)	262	8 MB	12	**4550**	**3660**	4270
740-2070 (1513)	262	8 MB	12	**4550**	**4550**	5308.3

C.22.2 Model 170 Servers

Current 170 Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

Table C.22.2.1 Current Model 170 Servers

Feature #	CPUs	Chip Speed	L2 cache per CPU	**Processor CPW**	**Interactive CPW (Knee)**	Interactive CPW (Max)	Processor CPU % @ Knee	Interactive CPU % @ Knee	Interactive CPU % @ Max
2289	1	200 MHz	n/a	**50**	**15**	17.5	70	30	35
2290	1	200 MHz	n/a	**73**	**20**	23.3	72.6	27.4	32
2291	1	200 MHz	n/a	**115**	**25**	29.2	78.3	21.7	25.4
2292	1	200 MHz	n/a	**220**	**30**	35	86.4	13.6	15.9
2385	1	252 MHz	4 MB	**460**	**50**	58.3	89.1	10.9	12.7
2386	1	252 MHz	4 MB	**460**	**70**	81.7	84.8	15.2	17.8
2388	2	255 MHz	4 MB	**1090**	**70**	81.7	92.3	6.4	7.5

Note: the CPU not used by the interactive workloads at their Max CPW is used by the system CFINTnn jobs. For example, for the 2386 model the interactive workloads use 17.8% of the CPU at their maximum and the CFINTnn jobs use the remaining 82.2%. The processor workloads use 0% CPU when the interactive workloads are using their maximum value.

AS/400e Dedicated Server for Domino

Table C.22.2.2 Dedicated Server for Domino

Feature #	CPUs	Chip Speed	L2 cache per CPU	**Processor CPW**	**Interactive CPW**	Processor CPU% @ Knee	Processor CPU % @ Max	Interactive CPU % @ Knee	Interactive CPU % @ Max
2407	1	n/a	n/a	30	10	-	-	-	-
2408	1	n/a	4 MB	60	15	-	-	-	-
2409	2	n/a	4 MB	120	20	-	-	-	-

Previous Model 170 Servers

On previous Model 170's the knee of the curve is about 1/3 the maximum interactive CPW value.

Note that a constrained (c) CPW rating means the maximum memory or DASD configuration is the constraining factor, not the processor. An unconstrained (u) CPW rating means the processor is the first constrained resource.

Table C.22.2.3 Previous Model 170 Servers

Feature #	Constrain / Unconstr	**Client / Server CPW**	**Interactive CPW (Max)**	Interactive CPW (Knee)	Interactive CPU % @ Max	Interactive CPU % @ Knee
2159	c	**73**	**16**	5.3	22.2	7.7
	u	**73**	**16**	5.3	22.2	7.7
2160	c	**114**	**23**	7.7	21.2	7.4
	u	**114**	**23**	7.7	21.2	7.4
2164	c	**125**	**29**	9.7	14	4.7
	u	**210**	**29**	9.7	14	4.7
2176	c	**125**	**40**	13.3	12.9	4.4
	u	**319**	**40**	13.3	12.9	4.4
2183	c	**125**	**67**	22.3	21.5	7.2
	u	**319**	**67**	22.3	21.5	7.2

C.23 AS/400e Model Sxx Servers

For AS/400e servers the knee of the curve is about 1/3 the maximum interactive CPW value.

Table C.23.1 AS/400e Servers

Model	Feature #	CPUs	**Max C/S CPW**	**Max Inter CPW**	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
S10	2118	1	**45.4**	**16.2**	5.4	35.7	11.9
	2119	1	**73.1**	**24.4**	8.1	33.4	11.1
S20	2161	1	**113.8**	**31**	10.3	27.2	9.1
	2163	1	**210**	**35.8**	11.9	17	5.7
	2165	2	**464.3**	**49.7**	16.7	10.7	3.6
	2166	4	**759**	**56.9**	19.0	7.5	2.5
S30	2257	1	**319**	**51.5**	17.2	16.1	5.4
	2258	2	**583.3**	**64**	21.3	11	3.7
	2259	4	**998.6**	**64**	21.3	6.4	2.1
	2260	8	**1794**	**64**	21.3	3.6	1.2
S40	2207	8	**3660**	**120**	40	3.2	1.1
	2208	12	**4550**	**120**	40	2.6	0.8
	2256	8	**1794**	**64**	21.3	3.6	1.2
	2261	12	**2340**	**64**	21.3	2.7	0.9

C.24 AS/400e Custom Servers

For custom servers the knee of the curve is about 6/7 maximum interactive CPW value.

Table C.24.1 AS/400e Custom Servers

Model	Feature #	CPUs	**Max**	**Max**	6/7 Max	CPU % @	CPU %
S20	2177	4	**759**	**110.7**	94.9	14.6	12.5
	2178	4	**759**	**221.4**	189.8	29.2	25.0
S30	2320	4	**998.6**	**215.1**	184.4	21.5	18.5
	2321	8	**1794**	**386.4**	331.2	21.5	18.5
	2322	8	**1794**	**579.6**	496.8	32.5	27.7
S40	2340	8	**3660**	**1050.0**	900.0	28.6	24.5
	2341	12	**4550**	**2050.0**	1757.1	38.6	33.1

C.25 AS/400 Advanced Servers

For AS/400 Advanced Servers the knee of the curve is about 1/3 the maximum interactive CPW value.

For releases prior to V4R1 the model 150 was constrained due to the memory capacity. With the larger capacity for V4R1, memory is no longer the limiting resource. In V4R1, the limit of 4 DASD devices is the constraining resource. For workloads that do not perform as many disk operations or don't require as much memory, the unconstrained CPW value may be more representative of the performance capabilities. An unconstrained CPW rating means the processor is the first constrained resource.

Table C.25.1 AS/400 Advanced Servers: V4R1 and V4R2								
Model	Feature #	Constrain / Unconstr	CPUs	**Max C/S CPW**	**Max Inter CPW**	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
150	2269	c	1	**20.2**	**13.8**	4.6	51.1	17
	2269	u	1	**27**	**13.8**	4.6	51.1	17
	2270	c	1	**20.2**	**20.2**	6.7	61.9	20.6
	2270	u	1	**35**	**20.6**	6.9	61.9	20.6
40S	2109	n/a	1	**27**	**9.4**	3.1	30.1	10
	2110	n/a	1	**35**	**14.5**	3.9	37.4	12.5
50S	2111	n/a	1	**63.0**	**21.6**	7.2	29.8	9.9
	2112	n/a	1	**91.0**	**32.2**	10.8	29.8	9.9
	2120	n/a	1	**81.6**	**22.5**	8.1	27.8	9.3
	2121	n/a	1	**111.5**	**32.2**	10.7	30	10
	2122	n/a	1	**138.0**	**32.2**	12.0	23.8	8.9
53S	2154	n/a	1	**188.2**	**32.2**	15.9	20.3	6.8
	2155	n/a	2	**319.0**	**32.2**	10.7	13.5	4.5
	2156	n/a	4	**598.0**	**32.2**	10.7	9	3
	2157	n/a	4	**650.0**	**32.2**	10.9	7.7	2.6

Table C.25.2 AS/400 Advanced Servers: V3R7								
Model	Feature #	Constrain / Unconstr	CPUs	**Max C/S CPW**	**Max Inter CPW**	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
150	2269	c	1	**10.9**	**10.9**	3.6	100.0	33.0
	2269	u	1	**10.9**	**10.9**	3.6	100.0	33.0
	2270	c	1	**27.0**	**13.8**	4.6	51.1	17.0
	2270	u	1	**33.3**	**20.6**	6.9	61.9	20.6
40S	2109	n/a	1	**27.0**	**9.4**	3.1	30.1	10
	2110	n/a	1	**33.3**	**13.8**	3.7	37.4	12.5
	2111	n/a	1	**59.8**	**20.6**	6.9	29.8	9.9
	2112	n/a	1	**87.3**	**30.7**	10.3	29.8	9.9
50S	2120	n/a	1	**77.7**	**21.4**	7.7	27.8	9.3
	2121	n/a	1	**104.2**	**30.7**	10.2	30	10
	2122	n/a	1	**130.7**	**30.7**	11.5	23.8	8.9
53S	2154	n/a	1	**162.7**	**30.7**	13.3	20.3	6.8
	2155	n/a	2	**278.8**	**30.7**	10.2	13.5	4.5
	2156	n/a	4	**459.3**	**30.7**	10.2	9	3
	2157	n/a	4	**509.9**	**30.7**	10.4	7.7	2.6

C.26 AS/400e Custom Application Server Model SB1

AS/400e application servers are particularly suited for environments with minimal database needs, minimal disk storage needs, lots of low-cost memory, high-speed connectivity to a database server, and minimal upgrade importance.

The throughput rates for Financial (FI) dialogsteps (ds) per hour may be used to size systems for customer orders. **Note: 1 SD ds = = 2.5 Fl ds**. (SD = Sales & Distribution).

Table C.26.1 AS/400e Custom Application Server Model SB1				
Model	**CPUs**	**SAP Release**	**SD ds/hr @ 65% CPU Utilization**	**FI ds/hr @ 65% CPU Utilization**
2312	8	3.1H	109,770.49	274,426.23
		4.0B	65,862.29	164,655.74
2313	12	3.1H	158,715.76	396,789.40
		4.0B	95,229.46	238,073.64

C.27 AS/400 Models 4xx, 5xx and 6xx Systems

Table C.27.1 AS/400 RISC Systems						
Model	**Feature Code**	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**V3R7 CPW**	**V4R1 CPW**
400	2130	1	160	50	13.8	13.8
	2131	1	224	50	20.6	20.6
	2132	1	224	50	27	27
	2133	1	224	50	33.3	35
500	2140	1	768	652	21.4	21.4
	2141	1	768	652	30.7	30.7
	2142	1	1024	652	43.9	43.9
510	2143	1	1024	652	77.7	81.6
	2144	1	1024	652	104.2	111.5
530	2150	1	4096	996	131.1	148
	2151	1	4096	996	162.7	188.2
	2152	2	4096	996	278.8	319
	2153	4	4096	996	459.3	598
	2162	4	4096	996	509.9	650

Table C.27.2 AS/400e Systems					
Model	**Feature Code**	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**V4R3 CPW**
600	2129	1	384	175.4	22.7
	2134	1	384	175.4	32.5
	2135	1	384	175.4	45.4
	2136	1	512	175.4	73.1
620	2175	1	1856	944.8	50
	2179	1	2048	944.8	85.6
	2180	1	2048	944.8	113.8
	2181	1	2048	944.8	210
	2182	2	4096	944.8	464.3
640	2237	1	16384	1340	319
	2238	2	8704	1340	583.3
	2239	4	16384	1340	998.6
650	2188	8	40960	2095.9	3660
	2189	12	40960	2095.9	4550
	2240	8	32768	2095.9	1794
	2243	12	32768	2095.9	2340

C.28 AS/400 CISC Model Capacities

Table C.28.1 AS/400 CISC Model: 9401					
Model	**Feature**	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**CPW**
P02	n/a	1	16	2.1	7.3
P03	2114	1	24	2.99	7.3
	2115	1	40	3.93	9.6
	2117	1	56	3.93	16.8

Table C.28.2 AS/400 CISC Model: 9402 Systems				
Model	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**CPW**
C04	1	12	1.3	3.1
C06	1	16	1.3	3.6
D02	1	16	1.2	3.8
D04	1	16	1.6	4.4
E02	1	24	2.0	4.5
D06	1	20	1.6	5.5
E04	1	24	4.0	5.5
F02	1	24	2.1	5.5
F04	1	24	4.1	7.3
E06	1	40	7.9	7.3
F06	1	40	8.2	9.6

Table C.28.3 AS/400 CISC Model: 9402 Servers					
Feature Code	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**C/S CPW**	**Interactive CPW**
S01	1	56	3.9	17.1	5.5
100	1	56	7.9	17.1	5.5

Table C.28.4 AS/400 CISC Model: 9404 Systems				
Model	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**CPW**
B10	1	16	1.9	2.9
C10	1	20	1.9	3.9
B20	1	28	3.8	5.1
C20	1	32	3.8	5.3
D10	1	32	4.8	5.3
C25	1	40	3.8	6.1
D20	1	40	4.8	6.8
E10	1	40	19.7	7.6
D25	1	64	6.4	9.7
F10	1	72	20.6	9.6
E20	1	72	19.7	9.7
F20	1	80	20.6	11.6
E25	1	80	19.7	11.8
F25	1	80	20.6	13.7

Table C.28.5 AS/400 CISC Model: 9404 Servers					
Feature Code	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**C/S CPW**	**Interactive CPW**
135	1	384	27.5	32.3	9.6
140	2	512	47.2	65.6	11.6

Table C.28.6 AS/400 CISC Model: 9406 Systems				
Model	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**CPW**
B30	1	36	13.7	3.8
B35	1	40	13.7	4.6
B40	1	40	13.7	5.2
B45	1	40	13.7	6.5
D35	1	72	67.0	7.4
B50	1	48	27.4	9.3
E35	1	72	67.0	9.7
D45	1	80	67.0	10.8
D50	1	128	98.0	13.3
E45	1	80	67.0	13.8
F35	1	80	67.0	13.7
B60	1	96	54.8	15.1
F45	1	80	67.0	17.1
E50	1	128	98.0	18.1
B70	1	192	54.8	20.0
D60	1	192	146	23.9
F50	1	192	114	27.8
E60	1	192	146	28.1
D70	1	256	146	32.3
E70	1	256	146	39.2
F60	1	384	146	40.0
D80	2	384	256	56.6
F70	1	512	256	57.0
E80	2	512	256	69.4
E90	3	1024	256	96.7
F80	2	768	256	97.1
E95	4	1152	256	116.6
F90	3	1024	256	127.7
F95	4	1280	256	148.8
F97	4	1536	256	177.4

Table C.28.7 AS/400 Advanced Systems (CISC)					
Model	**Feature Code**	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**CPW**
200	2030	1	24	23.6	7.3
	2031	1	56	23.6	11.6
	2032	1	128	23.6	16.8
300	2040	1	72	117.4	11.6
	2041	1	80	117.4	16.8
	2042	1	160	117.4	21.1
310	2043	1	832	159.3	33.8
	2044	2	832	159.3	56.5
320	2050	1	1536	259.6	67.5
	2051	2	1536	259.6	120.3
	2052	4	1536	259.6	177.4

Table C.28.8 AS/400 Advanced Servers (CISC)						
Model	**Feature Code**	**CPUs**	**Memory (MB) Maximum**	**Disk (GB) Maximum**	**C/S CPW**	**Interactive CPW**
20S	2010	1	128	23.6	17.1	5.5
2FS	2010	1	128	7.8	17.1	5.5
2SG	2010	1	128	7.8	17.1	5.5
2SS	2010	1	128	7.8	17.1	5.5
30S	2411	1	384	86.5	32.3	9.6
	2412	2	832	86.5	68.5	11.6