# iSeries Performance Capabilities Reference
# Version 5, Release 1

**October 2001**

This document is intended for use by qualified performance related programmers or analysts from IBM, IBM Business Partners and IBM customers using iSeries. Information in this document may be readily shared with IBM iSeries customers to understand the performance and tuning factors in OS/400 Version 5 Release 1. Requests for use of performance information by the technical trade press or consultants should be directed to Systems Performance Department V3T, IBM Rochester Lab, in Rochester, MN. 55901 USA.

**Note!**

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

**Sixteenth Edition (October 2001) SC41-0607-04**

This edition applies to Version 5, Release 1 of the AS/400 Operating System and iSeries platform

You can request a copy of this document by download from iSeries Information Center (On-line Library) via the iSeries Internet site at: http://www.ibm.com/eserver/iseries/ . The Version 4 Release 5 and Release 4 Performance Capabilities Guide are also available on the IBM iSeries Internet site in the "On Line Library", at: http://publib.boulder.ibm.com/pubs/html/as400/online/chgfrm.htm . Documents are viewable/downloadable in Adobe Acrobat (.pdf) format. Approximately 1 to 2 MB download. Adobe Acrobate reader plug-in is available at: http://www.adobe.com .

To request the CISC version (V3R2 and earlier), enter the following command on VM:

**REQUEST V3R2 FROM FIELDSIT AT RCHVMW2 (your name**

To request the IBM iSeries Advanced 36 version, enter the following command on VM:

**TOOLCAT MKTTOOLS GET AS4ADV36 PACKAGE**

# Contents

# Special Notices

**DISCLAIMER NOTICE**

Performance data in this document was obtained in a controlled environment with specific performance benchmarks and tools. This information is presented along with general recommendations to assist the reader to have a better understanding of IBM(*) products. Results obtained in other environments may vary significantly and does not predict a specific customer's environment.

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

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.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. 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.

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/400e | System/370 | Operating System/400 |
| C/400 | iSeries | iSeries |
| OS/400 | COBOL/400 | Application System/400 |
| PS/2 | RPG/400 | OfficeVision |
| OS/2 | CallPath | Facsimile Support/400 |
| DB2 | DRDA | Distributed Relational Database Architecture |
| AFP | SQL/400 | Advanced Function Printing |
| IBM | ImagePlus | Operational Assistant |
| SQL/DS | VTAM | Client Series |
| 400 | APPN | Workstation Remote IPL/400 |
| CICS | SystemView | Advanced Peer-to-Peer Networking |
| S/370 | ValuePoint | OfficeVision/400 |
| RPG IV | DB2/400 | iSeries Advanced Application Architecture |
| AIX | ADSM/400 | ADSTAR Distributed Storage Manager/400 |
| IPDS | AnyNet/400 | IBM Network Station |

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 |
| Lotus Notes, Lotus, Word Pro | Lotus Development Corporation |
| Notes, 123, CC Mail, Freelance | Lotus Development Corporation |
| Microsoft, Windows 95 | Microsoft Corporation |
| Windows 95, Windows 95 Explorer | Microsoft Corporation |
| Microsoft Word, PowerPoint, Excel | Microsoft Corporation |
| 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 |
| dBASEIII Plus | Borland International |
| 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. |
| Pentium | Intel Corporation |
| SPEC | Syems 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 |
| Java | Sun Microsystems, Inc. |

Other terms that are used in this document may be trademarks of other companies.

## Purpose of this Document

**The intent of this document is to provide guidance in terms of iSeries performance, capacity planning information, and tips to obtain best performance.** This document is typically updated with each new release or more often if needed. This **October 2001 edition** of the V5R1 Performance Capabilities Reference Guide is an update to the April 2001edition to reflect new product functions announce on April 23, 2001, with available through the remainder of 2001. The April 2001 edition superceeds the V4R5 July 2000 edition.

This October 2001 Edition include new information on MQ Series, WebSphere 4.0, external disk, security, PM/400 and TCP/IP performance.

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 Workload Estimator for iSeries or BEST/1(**) for OS/400 (refer to Appendix B for more details on BEST/1 support).

### Related Publications / Documents

The following publications/documents are considered particularly suitable for additional information on iSeries performance topics.

- *iSeries Programming: Work Management Guide*, SC41-4306
- *iSeries System Handbook, GA19-5486*
- *iSeries Programming: Performance Tools/400 Guide*, SC41-8084

# Chapter 1. Introduction

V5R1 continues to enhance the iSeries value proposition - the best melding of a superior operating system with 64-bit RISC hardware. The performance of V5R1 is greatly improved by both software enhancements and new hardware. iSeries continues to deliver customer usable performance in the multi-user, multi-applications environment by supporting interactive, client server, batch, groupware (Domino), Java, business intelligence, and web (WebSphere) serving.

High-end 24-way processor was enhanced in V5R1, increasing performance of iSeries by 20+% over that available in V4R5. V5R1 supports new Model 820 and 840 servers, new Model 270s, new Base models and new Dedicated Servers for Domino. In addition, the new models offer interactive performance features for the new server/interactive processing flexibility. These new models offer better compute intensive, interactive/server mode performance and significant price/performance improvements. There is now over a 300-fold range of scalability in performance from the smallest iSeries model to the largest 24-way.

The primary V5R1 extreme performance items are:

- New models for high-end growth, with extreme scaling up to 24-way processors with up to 20,200 CPW (Commercial Processing Workload) and 10,950 CIW (Compute Intentive Workload) metrics
- Improved V5R1 CPW values, extremely fast processors (up to 600MHz), large L2 caches (up to 16MB), more memory (up to 131GB)
- Enhanced server model performance characteristics and interactive/server algorithm
- Improved NT performance on Integrated xSeries™ Server for iSeries (850 MHz processor) and support for Windows 2000 via Integrated xSeries Server Adapter.
- Reduced storage cost with possible performance improvements using first-to-market integrated hardware data compression (IHDC) for both disk and tape compression along with a new bus and memory cross-bar switch of up to 42GB per second (up to 12X performance improvement)
- Support for 10k rpm disks, 1.6GB Read Cache and IBM Shark Storage Server for sharing disks
- Logical Partitioning (LPAR) of OS/400 for multiple simultaneous partitions (up to 32) with separate processors, storage, clock, primary language and currency capabilities
- Universal Database support for new image, video, audio and other larger object types in DB2/400
- Enhanced support for continuous availability clustering and disk swapping
- New secure Enterprise-class TCP/IP support and dramatic Gigabit Ethernet performance improvements
- Improved Lotus Domino mail and calendar performance with up to 77,800 Mail/Calendar Users and new Dedicated Servers for Domino with even better price/performance
- Dramatic server-side Java performance measured by VolanoMark and SPECjBB2000 benchmarks
- Parallel save/restore performance improvements with hierarchical storage management support based on user-defined policies and parallel single object support for multiple tapes up to 2.2TB per hour
- WebSphere and secure web server performance improvements
- Portable Application Solutions Environment (PASE) with new AIX 4.3 64-bit application execution
- Linux supported in secondary partitions of LPAR

Customers who wish to remain with their existing hardware but want to move to the V5R1 operating system may find functional and performance improvements. Version 5 Release 1 OS/400 continues to protect the customer's investment while providing more function, growth, capacity, performance and better price/performance over previous versions. V5R1 operating system will run on all previous RISC models. Primary public performance website is found at: http://www.ibm.com/eserver/iseries/perfmgmt/ .

# Chapter 2. iSeries and AS/400 RISC Server Model Performance Behavior

## 2.1 Overview

iSeries 400, AS/400* Advanced Servers, and AS/400e* servers are intended for use primarily in client/server or other non-interactive work environments such as batch, business intelligence, network computing etc. 5250-based interactive work can be run on these servers, but with limitations. With iSeries and AS/400 servers, interactive capacity can be increased with the purchase of additional interactive features. Interactive work is defined as any job doing 5250 display device I/O. This includes:

| | |
|---|---|
| All 5250 sessions | RUMBA/400 |
| Any green screen interface | Screen scrapers |
| Telnet or 5250 DSPT workstations | Interactive subsystem monitors |
| 5250/HTML workstation gateway | Twinax printer jobs |
| PC's using 5250 emulation | BSC 3270 emulation |
| Interactive program debugging | 5250 emulation |
| PC Support/400 work station function | |

There are cases when printer work can be considered to be interactive, but these cases are rare. For example, pressing the "Print" key on a workstation that has a directly connected printer will cause interactive work to occur. Also, if an interactive job allocates a printer and bypasses the spooling subsystem, printing associated with that job will be interactive. Normal print operations that use spooling (QSPL subsystem) with writers processing outqueues are not considered to be interactive work.

### 2.1.1 Interactive Indicators and Metrics

Prior to V4R5, there were no system metrics that would allow a customer to determine the overall interactive feature capacity utilization. It was difficult for the customer to determine how much of the total interactive capacity he was using and which jobs were consuming interactive capacity. This got much easier with the system enhancements made in V4R5 and V5R1
.
Starting with V4R5, a new metric was added to the data generated by collection services to report the system's interactive CPU utilization (ref file QAPMSYSCPU). Also, interactive feature utilization was reported when printing a System Report generated from collection services data . In addition, Management Central now monitors interactive CPU relative to the system/partition capacity.

New in V5R1, collection services was enhanced to mark all tasks that are counted against interactive capacity (ref file QAPMJOBMI, field JBSVIF set to ‘1’). It is possible to query this file and get the CPU utilized for all interactive tasks.

With the above enhancements, a customer can easily monitor the usage of interactive feature and decide when he is approaching the need for an interactive feature upgrade.

### 2.1.2 Disclaimer and Remaining Sections

The performance information and equations in this chapter represent ideal environments. This information is presented along with general recommendations to assist the reader to have a better understanding of the AS/400 server models. Actual results may vary significantly.

This chapter is organized into the following sections:

- Server Model Behavior
- Server Model Differences
- Performance Highlights of New Model 7xx Servers
- Performance Highlights of Current Model 170 Servers
- Performance Highlights of Custom Server Models
- Additional Server Considerations
- Interactive Utilization
- Server Dynamic Tuning (SDT)
- Managing Interactive Capacity
- Migration from Traditional Models
- Migration from Server Models
- AS/400e Dedicated Server for Domino (DSD) Performance Behavior

### 2.1.3 V5R1 Additions

There were several new iSeries 400 8xx and 270 server model additions in V5R1. However, with the exception of the DSD models, the underlying server behavior did not change from V4R5. For more details on these new models, please refer to *Appendix D, iSeries CPW Values*.

Five new iSeries 400 DSD models are introduced with V5R1. In addition, V5R1 expands the capability of the DSD models with enhanced support of Domino-complementary workloads such as Java Servlets and WebSphere Application Server. Please refer to Section 2.13, *Dedicated Server for Domino Performance Behavior*, for additional information.

## 2.2 Server Model Behavior

### 2.2.1 New in V4R5

In V4R5 there are several new 2xx, 8xx and SBx model servers. These new servers models utilize an enhanced server algorithm that manages the interactive CPU utilization. This enhanced server algorithm may provide significant user benefit. On prior models, when interactive users exceed the interactive CPW capacity of a system, additional CPU usage visible in one or more CFINT tasks, reduces system capacity for all users including client/server. New in V4R5, the system attempts to hold interactive CPU utilization below the threshold where CFINT CPU usage begins to increase. Only in extreme cases, where interactive demand exceeds the limitations of the interactive capacity for an extended time (usually from long-running, CPU-intensive transactions), will overhead be visable via the CFINT tasks. Highlights of this new algorithm include the following:

- As interactive users exceed the installed interactive CPW capacity, the response times of those applications may significantly lengthen and the system will attempt to manage these interactive excesses below a level where CFINT CPU usage begins to increase. Generally, increased CFINT may still occur but only for brief transient periods of time. Therefore, there should be remaining system capacity available for non-interactive users of the system even though the interactive capacity has been exceeded. It is still a good practice to keep interactive system use below the system interactive CPW threshold to avoid long interactive response times.
- Client/server users should be able to utilize most of the remaining system capacity even though the interactive users have temporarily exceeded the maximum interactive CPW capacity.
- The new AS/400e Dedicated Server for Domino models behave similarly when the Non Domino CPW capacity has been exceeded (i.e. the system attempts to hold Non Domino CPW capacity below the threshold where CFINT overhead is normally activated). Thus, Domino users should be able to run in the remaining system capacity available.
- With the advent of the new server algorithm, there is not a concept known as the interactive knee or interactive cap. The system just attempts to manage the interactive CPU utilization to the level of the interactive CPW capacity.
- Dynamic priority adjustment (system value QDYNPTYADJ) will not have any effect managing the interactive workloads as they exceed the system interactive CPW capacity. On the other hand, it won't hurt to have it activated.
- The new server algorithm only applies to the new hardware available in V4R5 (2xx, 8xx and SBx models) . The behavior of all other hardware, such as the 7xx models is unchanged.

### 2.2.2 Choosing Between Similarly Rated Systems

Sometimes it is necessary to choose between two systems that have similar CPW values but different processor megahertz (MHz) values or L2 cache sizes. If your applications tends to be compute intensive such as Java, WebSphere, EJBs, and Domino, you may want to go with the faster MHz processors because you will generally get faster response times. However, if your response times are already sub-second, it is not likely that you will notice the response time improvements. If your applications tend to be L2 cache friendly such as many traditional commercial applications are, you may want to choose the system that has the larger L2 cache. In either case, you can use the IBM Workload Estimator for AS/400 to help you select the correct system (see URL: http://as400service.ibm.com/estimator ) .

### 2.2.3 Existing Models

Server model behavior applies to:

- AS/400 Advanced Servers
- AS/400e servers
- AS/400e custom servers
- AS/400e model 150
- AS/400e model 170
- AS/400e model 7xx

Relative performance measurements are derived from commercial processing workload (CPW) on AS/400. CPW is representative of commercial applications, particularly those that do significant database processing in conjunction with journaling and commitment control.

Traditional (non-server) AS/400 system models had a single CPW value which represented the maximum workload that can be applied to that model. This CPW value was applicable to either an interactive workload, a client/server workload, or a combination of the two.

Now there are two CPW values. The larger value represents the maximum workload the model could support if the workload were entirely client/server (i.e. no interactive components). This CPW value is for the processor feature of the system. The smaller CPW value represents the maximum workload the model could support if the workload were entirely interactive. For 7xx models this is the CPW value for the interactive feature of the system.

**The two CPW values are NOT additive - interactive processing will reduce the system's client/server processing capability.** When 100% of client/server CPW is being used, there is no CPU available for interactive workloads. When 100% of interactive capacity is being used, there is no CPU available for client/server workloads.

For model 170s announced in 9/98 and all subsequent systems, the published interactive CPW represents the point (the "knee of the curve") where the interactive utilization may cause increased overhead on the system. (As will be discussed later, this threshold point (or knee) is at a different value for previously announced server models.) Up to the knee the server/batch capacity is equal to the processor capacity (CPW) minus the interactive workload. As interactive requirements grow beyond the knee, overhead grows at a rate which can eventually eliminate server/batch capacity and limit additional interactive growth. **It is best for interactive workloads to execute below (less than) the knee of the curve.** (However, for those models having the knee at 1/3 of the total interactive capacity, satisfactory performance can be achieved.) The following graph illustrates these points.

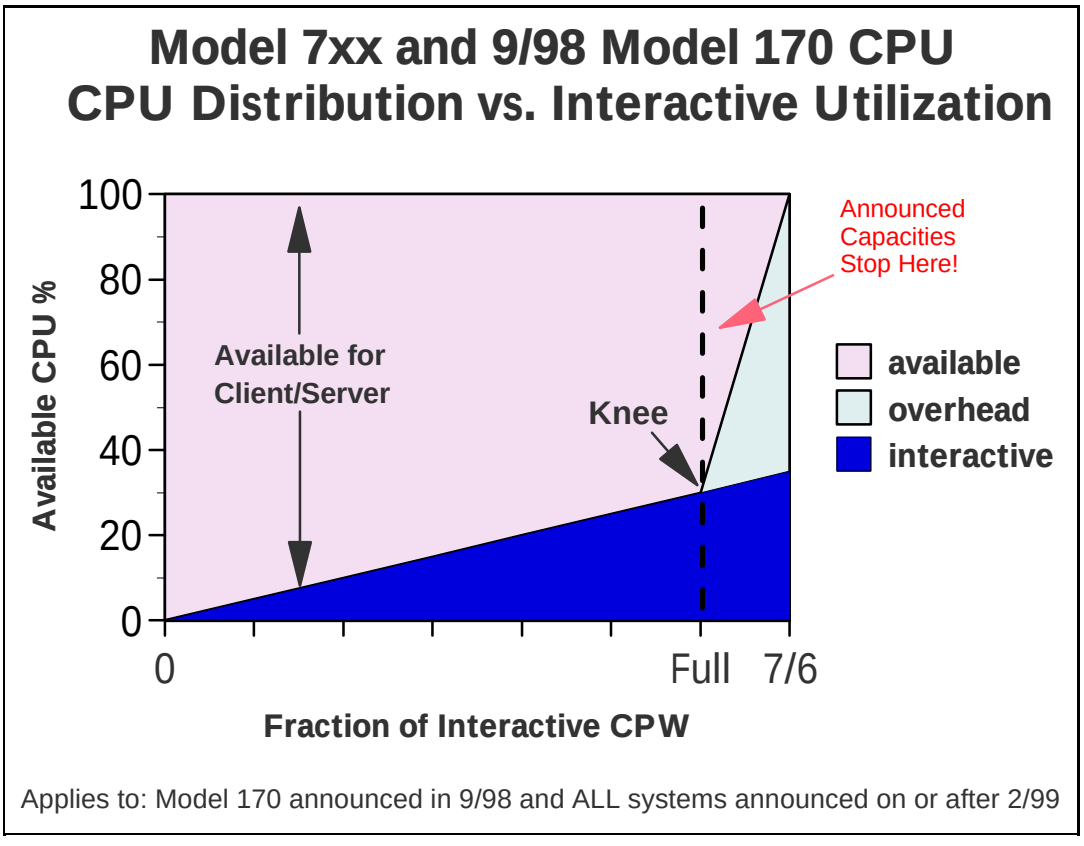


*Figure 2.1. Server Model behavior*

The figure above shows a straight line for the effective interactive utilization. Real/customer environments will produce a curved line since most environments will be dynamic, due to job initiation, interrupts, etc.

In general, a single interactive job will not cause a significant impact to client/server performance

Microcode task CFINTn, for all AS/400 models, handles interrupts, task switching, and other similar system overhead functions. For the server models, when interactive processing exceeds a threshold amount, the additional overhead required will be manifest in the CFINTn task. Note that a single interactive job will not incur this overhead.

There is one CFINTn task for each processor. For example, on a single processor system only CFINT1 will appear. On an 8-way processor, system tasks CFINT1 through CFINT8 will appear. It is possible to see significant CFINT activity even when server/interactive overhead does not exist. For example if there are lots of synchronous or communication I/O or many jobs with many task switches.

The effective interactive utilization (EIU) for a server system can be defined as the useable interactive utilization plus the total of CFINT utilization.

---

## 2.3 Server Model Differences

Server models were designed for a client/server workload and to accommodate an interactive workload. When the interactive workload exceeds an interactive CPW threshold (the “knee of the curve”) the client/server processing performance of the system becomes increasingly impacted at an accelerating rate beyond the knee as interactive workload continues to build. Once the interactive workload reaches the maximum interactive CPW value, all the CPU cycles are being used and there is no capacity available for handling client/server tasks.

Custom server models interact with batch and interactive workloads similar to the server models but the degree of interaction and priority of workloads follows a different algorithm and hence the knee of the curve for workload interaction is at a different point which offers a much higher interactive workload capability compared to the standard server models.

For the server models the knee of the curve is approximately:

- 100% of interactive CPW for:
  - AS/400e model 170s announced on or after 9/98
  - 7xx models
- 6/7 (86%) of interactive CPW for:
  - AS/400e custom servers
- 1/3 of interactive CPW for:
  - AS/400 Advanced Servers
  - AS/400e servers
  - AS/400e model 150
  - AS/400e model 170s announced in 2/98

For the 7xx models the interactive capacity is a feature that can be sized and purchased like any other feature of the system (i.e. disk, memory, communication lines, etc.).

The following charts show the CPU distribution vs. interactive utilization for Custom Server and pre-2/99 Server models.

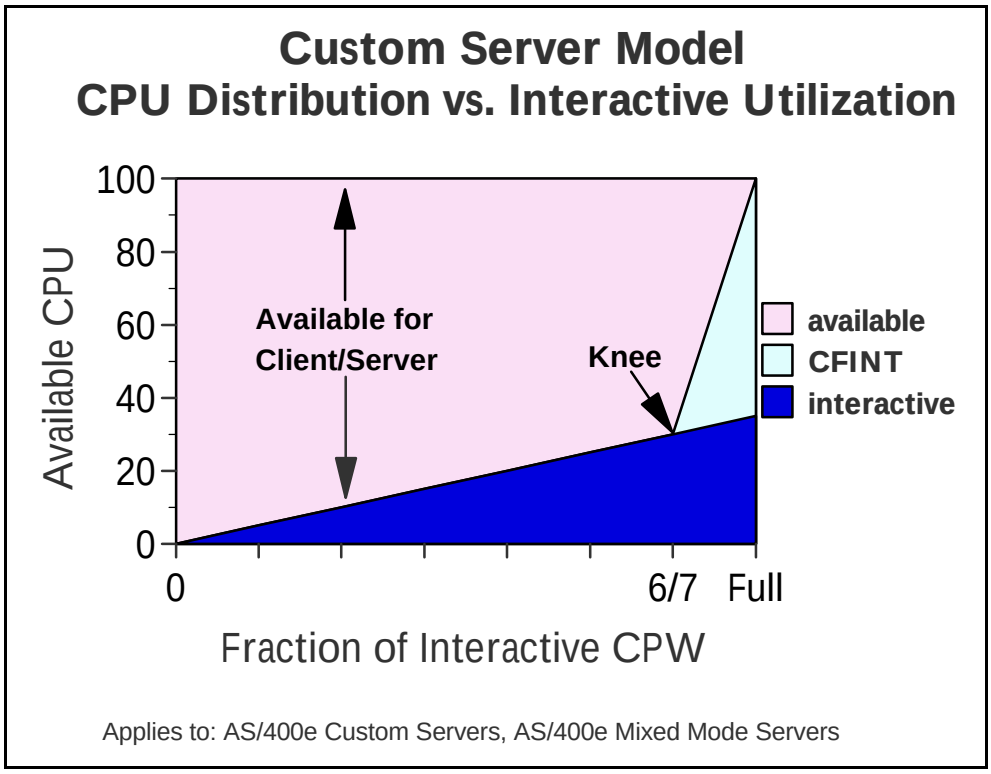



*Figure 2.2. Custom Server Model behavior*

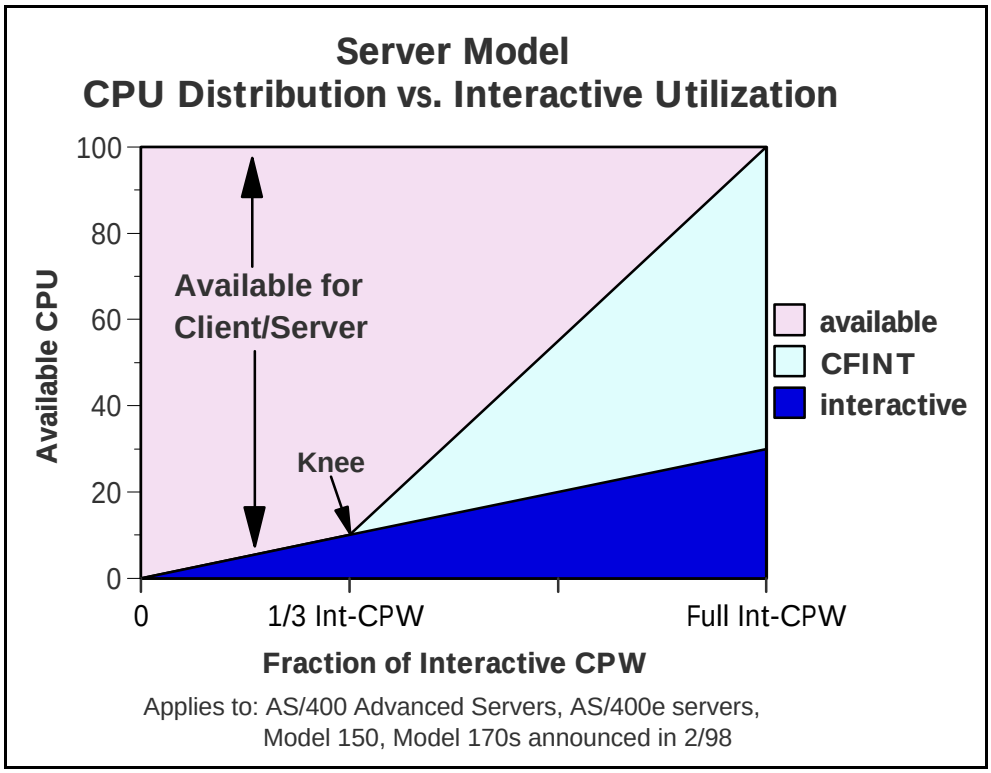



*Figure 2.3. Server Model behavior*

## 2.4 Performance Highlights of New Model 7xx Servers

7xx models were designed to accommodate a mixture of traditional "green screen" applications and more intensive "server" environments. Interactive features may be upgraded if additional interactive capacity is required. This is similar to disk, memory, or other features.

Each system is rated with a **processor CPW** which represents the relative performance (maximum capacity) of a processor feature running a commercial processing workload (CPW) in a client/server environment. **Processor CPW** is achievable when the commercial workload is not constrained by main storage or DASD.

Each system may have one of several interactive features. Each interactive feature has an **interactive CPW** associated with it. **Interactive CPW** represents the relative performance available to perform host-centric (5250) workloads. The amount of interactive capacity consumed will reduce the available processor capacity by the same amount. The following example will illustrate this performance capacity interplay:

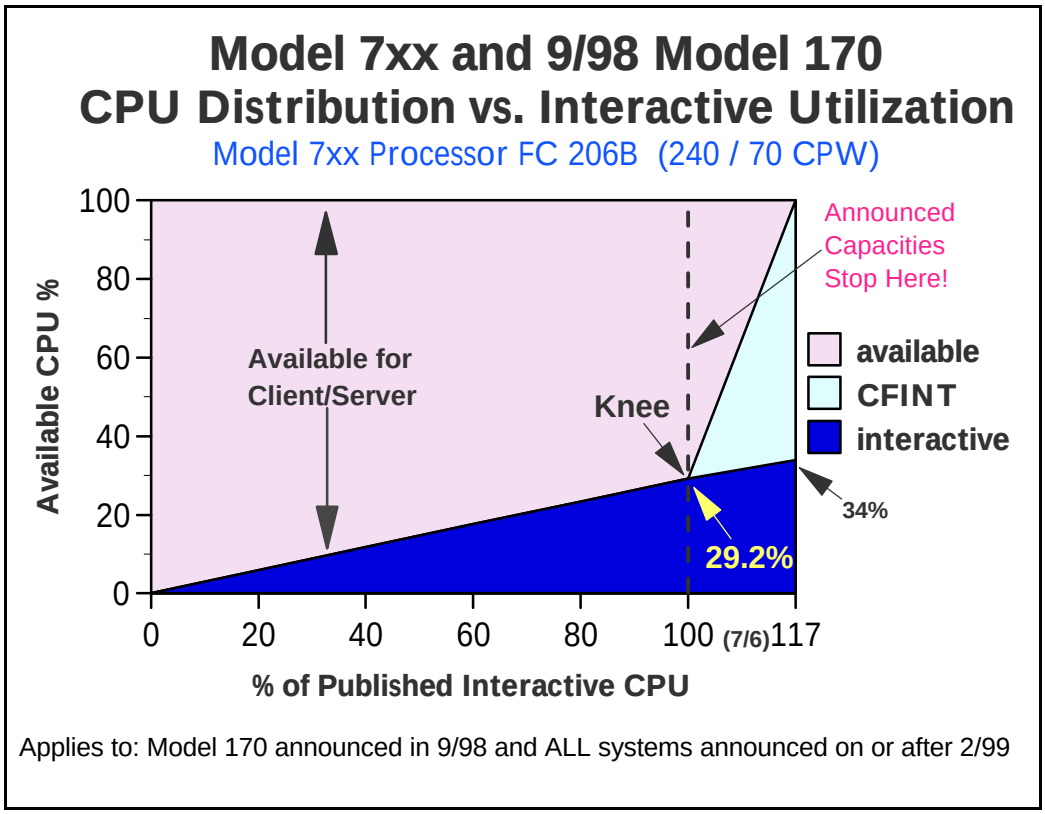



*Figure 2.4. Model 7xx Utilization Example*

At 110% of percent of the published interactive CPU, or 32.1% of total CPU, CFINT will use an additional 39.8% (approximate) of the total CPU, yielding an effective interactive CPU utilization of approximately 71.9%. This leaves approximately 28.1% of the total CPU available for client/server work. Note that the CPU is completely utilized once the interactive workload reaches about 34%. (CFINT would use approximately 66% CPU). At this saturation point, there is no CPU available for client/server.

## 2.5 Performance Highlights of Current Model 170 Servers

AS/400e Dedicated Server for Domino models will be generally available on September 24, 1999. Please refer to Section 2.13, *AS/400e Dedicated Server for Domino Performance Behavior*, for additional information.

Model 170 servers (features 2289, 2290, 2291, 2292, 2385, 2386 and 2388) are significantly more powerful than the previous Model 170s announced in Feb. '98. They have a faster processor (262MHz vs. 125MHz) and more main memory (up to 3.5GB vs. 1.0GB). In addition, the interactive workload balancing algorithm has been improved to provide a linear relationship between the client/server (batch) and published interactive workloads as measured by CPW.

The CPW rating for the maximum client/server workload now reflects the relative processor capacity rather than the "system capacity" and therefore there is no need to state a "constrained performance" CPW. This is because some workloads will be able to run at processor capacity if they are not DASD, memory, or otherwise limited.

Just like the model 7xx, the current model 170s have a **processor capacity** (CPW) value and an i**nteractive capacity** (CPW) value. These values behave in the same manner as described in the **Performance highlights of new model 7xx servers** section.

As interactive workload is added to the current model 170 servers, the remaining available client/server (batch) capacity available is calculated as: **CPW (C/S batch) = CPW(processor) - CPW(interactive)** This is valid up to the published interactive CPW rating. As long as the interactive CPW workload does not exceed the published interactive value, then interactive performance and client/server (batch) workloads will be both be optimized for best performance. **Bottom line, customers can use the entire interactive capacity with no impacts to client/server (batch) workload response times.**

On the current model 170s, if the **published interactive capacity** is exceeded, system overhead grows very quickly, and the client/server (batch) capacity is quickly reduced and becomes zero once the interactive workload reaches 7/6 of the published interactive CPW for that model.

The absolute limit for dedicated interactive capacity on the current models can be computed by multiplying the published interactive CPW rating by a factor of 7/6. The absolute limit for dedicated client/server (batch) is the published processor capacity value. This assumes that sufficient disk and memory as well as other system resources are available to fit the needs of the customer's programs, etc. Customer workloads that would require more than 10 disk arms for optimum performance should not be expected to give optimum performance on the model 170, as 10 disk access arms is the maximum configuration.

When the model 170 servers are running less than the published interactive workload, no Server Dynamic Tuning (SDT) is necessary to achieve balanced performance between interactive and client/server (batch) workloads. However, as with previous server models, a system value (QDYNPTYADJ - Server Dynamic Tuning ) is available to determine how the server will react to work requests when interactive workload exceeds the "knee". If the QDYNPTYADJ value is turned on, client/server work is favored over additional interactive work. If it is turned off, additional interactive work is allowed at the expense of low-priority client/server work. QDYNPTYADJ only affects the server when interactive requirements exceed the published interactive capacity rating. The shipped default value is for QDYNPTYADJ to be turned on.

The next chart shows the performance capacity of the current and previous Model 170 servers.

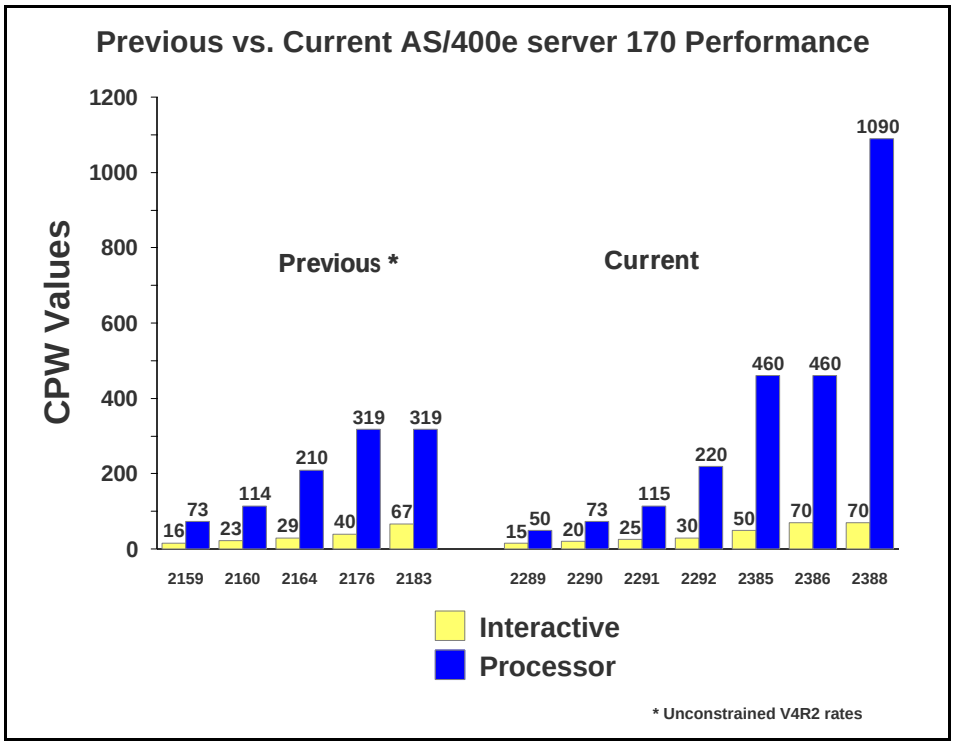



Figure 2.5. *Previous vs. Current Server 170 Performance*

## 2.6 Performance Highlights of Custom Server Models

Custom server models were available in releases V4R1 through V4R3. They interact with batch and interactive workloads similar to the server models but the degree of interaction and priority of workloads is different, and the knee of the curve for workload interaction is at a different point. When the interactive workload exceeds approximately 6/7 of the maximum interactive CPW (the knee of the curve), the client/server processing performance of the system becomes increasingly impacted. Once the interactive workload reaches the maximum interactive CPW value, all the CPU cycles are being used and there is no capacity available for handling client/server tasks.

## 2.7 Additional Server Considerations

It is recommended that the System Operator job run at runpty(9) or less. This is because the possibility exists that runaway interactive jobs will force server/interactive overhead to their maximum. At this point it is difficult to initiate a new job and one would need to be able to work with jobs to hold or cancel runaway jobs.

You should monitor the interactive activity closely. To do this take advantage of PM/400 or else run the Performance Monitor tool nearly continuously and query monitor data base each day for high interactive use and higher than normal CFINT values. The goal is to avoid exceeding the threshold (knee of the curve) value of interactive capacity.

## 2.8 Interactive Utilization

When the interactive CPW utilization is beyond the knee of the curve, the following formulas can be used to determine the effective interactive utilization or the available/remaining client/server CPW. ***These equations apply to all server models*.**

**CPWcs(maximum) =** client/server CPW maximum value
**CPWint(maximum) =** interactive CPW maximum value
**CPWint(knee) =** interactive CPW at the knee of the curve
**CPWint =** interactive CPW of the workload

**X** is the ratio that says how far into the overhead zone the workload has extended:
**X = (CPWint - CPWint(knee)) / (CPWint(maximum) - CPWint(knee))**

**EIU =** Effective interactive utilization**.** In other words, the free running**, CPWint(knee)**, interactive plus the combination of interactive and overhead generated by **X.**
**EIU = CPWint(knee) + (X * (CPWcs(maximum) - CPWint(knee)))**

**CPW remaining for batch = CPWcs(maximum) - EIU**

**Example 1:**

A model 7xx server has a Processor CPW of **240** and an Interactive CPW of **70**.
The interactive CPU percent at the knee equals (70 CPW / 240 CPW) or **29.2**%.
The maximum interactive CPU percent (7/6 of the Interactive CPW ) equals (81.7 CPW / 240 CPW) or **34**%.

Now if the interactive CPU is held to less than **29.2**% CPU (the knee), then the CPU available for the System, Batch, and Client/Server work is **100% - the Interactive CPU used**.

If the interactive CPU is allowed to grow above the knee, say for example **32.1** % (110% of the knee), then the CPU percent remaining for the Batch and System is calculated using the formulas above:
**X = (32.1 - 29.2) / (34 - 29.2) = .604**
**EIU = 29.2 + (.604 * (100 - 29.2)) = 71.9%**

**CPW remaining for batch = 100 - 71.9 = 28.1%**

Note that a swing of + or - 1% interactive CPU yields a swing of effective interactive utilization **(EIU)** from 57% to 87%. Also note that on custom servers and 7xx models, environments that go beyond the interactive knee may experience erratic behavior.

**Example 2:**

A Server Model has a Client/Server CPW of **450** and an Interactive CPW of **50**.
The maximum interactive CPU percent equals (50 CPW / 450 CPW) or **11**%.
The interactive CPU percent at the knee is 1/3 the maximum interactive value. This would equal **4**%.

Now if the interactive CPU is held to less than **4**% CPU (the knee), then the CPU available for the System, Batch, and Client/Server work is **100% - the Interactive CPU used**.

If the interactive CPU is allowed to grow above the knee, say for example **9**% (or 41 CPW), then the CPU percent remaining for the Batch and System is calculated using the formulas above:

**X = (9 - 4) / (11 - 4) = .71** (percent into the overhead area)
**EIU = 4 + (.71 * (100 - 4)) = 72%**

**CPW remaining for batch = 100 - 72 = 28%**

Note that a swing of + or - 1% interactive CPU yields a swing of effective interactive utilization **(EIU)** from 58% to 86%.

On earlier server models, the dynamics of the interactive workload beyond the knee is not as abrupt, but because there is typically less relative interactive capacity the overhead can still cause inconsistency in response times.

---

## 2.9 Server Dynamic Tuning (SDT)

Logic was added in V4R1 and is still in use today so customers could better control the impact of interactive work on their client/server performance. Note that with the new Model 170 servers (features 2289, 2290, 2291, 2292, 2385, 2386 and 2388) this logic only affects the server when interactive requirements exceed the published interactive capacity rating. For further details see the section, **Performance highlights of current model 170 servers**.

Through dynamic prioritization, all interactive jobs will be put lower in the priority queue, approximately at the knee of the curve. Placing the interactive jobs at a lesser priority causes the interactive jobs to slow down, and more processing power to be allocated to the client/server processing. As the interactive jobs receive less processing time, their impact on client/server processing will be lessened. When the interactive jobs are no longer impacting client/server jobs, their priority will dynamically be raised again.

The dynamic prioritization acts as a regulator which can help reduce the impact to client/server processing when additional interactive workload is placed on the system. In most cases, this results in better overall throughput when operating in a mixed client/server and interactive environment, but it can cause a noticeable slowdown in interactive response.

To fully enable SDT, customers MUST use a non-interactive job run priority (RUNPTY parameter) value of 35 or less (which raises the priority, closer to the default priority of 20 for interactive jobs).

Changing the existing non-interactive job's run priority can be done either through the Change Job (CHGJOB) command or by changing the RUNPTY value of the Class Description object used by the non-interactive job. This includes IBM-supplied or application provided class descriptions.

Examples of IBM-supplied class descriptions with a run priority value higher than 35 include QBATCH and QSNADS and QSYSCLS50. Customers should consider changing the RUNPTY value for QBATCH and QSNADS class descriptions or changing subsystem routing entries to not use class descriptions QBATCH, QSNADS, or QSYSCLS50.

If customers modify an IBM-supplied class description, they are responsible for ensuring the priority value is 35 or less after each new release or cumulative PTF package has been installed. One way to do this is to include the Change Class (CHGCLS) command in the system Start Up program.

NOTE: Several IBM-supplied class descriptions already have RUNPTY values of 35 or less. In these cases no user action is required. One example of this is class description QPWFSERVER with RUNPTY(20). This class description is used by Client Access database server jobs QZDAINIT (APPC) and QZDASOINIT (TCP/IP).

The system deprioritizes jobs according to groups or "bands" of RUNPTY values. For example, 10-16 is band 1, 17-22 is band 2, 23-35 is band 3, and so on.

Interactive jobs with priorities 10-16 are an exception case with V4R1. Their priorities will not be adjusted by SDT. These jobs will always run at their specified 10-16 priority.

When only a single interactive job is running, it will not be dynamically reprioritized.

When the interactive workload exceeds the knee of the curve, the priority of all interactive jobs is decreased one priority band, as defined by the Dynamic Priority Scheduler, every 15 seconds. If needed, the priority will be decreased to the 52-89 band. Then, if/when the interactive CPW work load falls below the knee, each interactive job's priority will gradually be reset to its starting value when the job is dispatched.

If the priority of non-interactive jobs are not set to 35 or lower, SDT stills works, but its effectiveness is greatly reduced, resulting in server behavior more like V3R6 and V3R7. That is, once the knee is exceeded, interactive priority is automatically decreased. Assuming non-interactive is set at priority 50, interactive could eventually get decreased to the 52-89 priority band. At this point, the processor is slowed and interactive and non-interactive are running at about the same priority. (There is little priority difference between 47-51 band and the 52-89 band.) If the Dynamic Priority Scheduler is turned off, SDT is also turned off.

Note that even with SDT, the underlying server behavior is unchanged. Customers get no more CPU cycles for either interactive or non-interactive jobs. SDT simply tries to regulate interactive jobs once they exceed the knee of the curve.

Obviously systems can still easily exceed the knee and stay above it, by having a large number of interactive jobs, by setting the priority of interactive jobs in the 10-16 range, by having a small client/server workload with a modest interactive workload, etc. The entire server behavior is a partnership with customers to give non-interactive jobs the bulk of the CPU while not entirely shutting out interactive.

To enable the Server Dynamic Tuning enhancement ensure the following system values are on:
(the shipped defaults are that they are set on)

- QDYNPTYSCD - this improves the job scheduling based on job impact on the system.
- QDYNPTYADJ - this uses the scheduling tool to shift interactive priorities after the threshold is reached.

The Server Dynamic Tuning enhancement is most effective if the batch and client/server priorities are in the range of 20 to 35.

***Server Dynamic Tuning Recommendations***

On the new systems and mixed mode servers have the QDYNPTYSCD and QDYNPTYADJ system value set on. This preserves non-interactive capacities and the interactive response times will be dynamic beyond the knee regardless of the setting. Also set non-interactive class run priorities to less than 35.

On earlier servers and 2/98 model 170 systems use your interactive requirements to determine the settings. For "pure interactive" environments turn the QDYNPTYADJ system value off. in mixed environments with important non-interactive work, leave the values on and change the run priority of important non-interactive work to be less than 35.

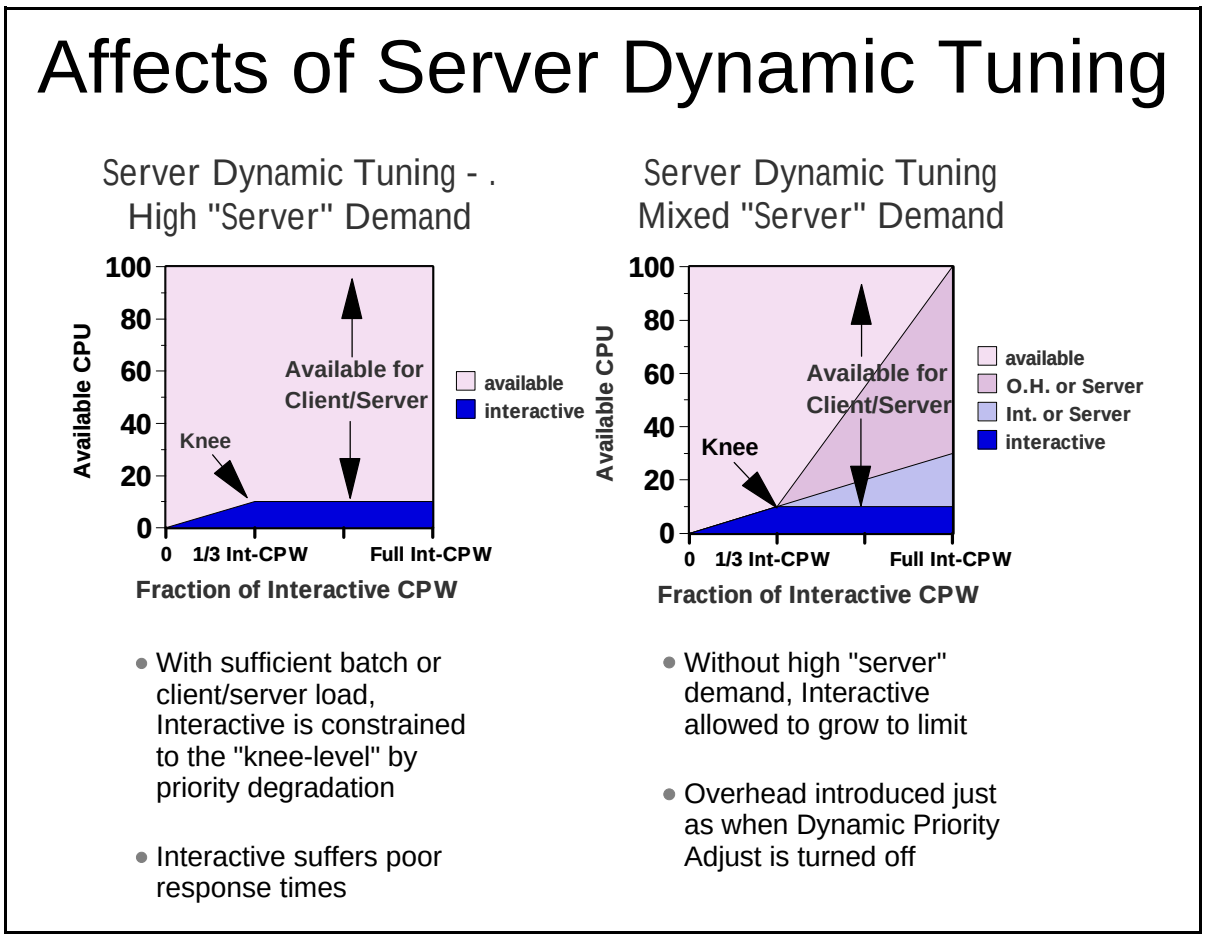



Figure 2.6.

## 2.10 Managing Interactive Capacity

***Interactive/Server characteristics in the real world.***

Graphs and formulas listed thus far work perfectly, provided the workload on the system is highly regular and steady in nature. Of course, very few systems have workloads like that. The more typical case is a dynamic combination of transaction types, user activity, and batch activity. There may very well be cases where the interactive activity exceeds the documented limits of the interactive capacity, yet decreases quickly enough so as not to seriously affect the response times for the rest of the workload. On the other hand, there may also be some intense transactions that force the interactive activity to exceed the documented limits interactive feature for a period of time even though the average CPU utilization appears to be less than these documented limits.

For 7xx systems, current 170 systems, and mixed-mode servers, a goal should be set to only rarely exceed the threshold value for interactive utilization. This will deliver the most consistent performance for both interactive and non-interactive work.

The questions that need to be answered are:
1. "How do I know whether my system is approaching the interactive limits or not?"
2. "What is viewed as 'interactive' by the system?"
3. "How close to the threshold can a system get without disrupting performance?"

This section attempts to answer these questions.

***Observing Interactive CPU utilization***

The most commonly available method for observing interactive utilization is the Performance Monitor used in conjunction with the Performance Tools program product. The monitor collects data for each job on the system, including the CPU consumed and the type of job. By examining the reports generated by the Performance Tools product, or by writing a query against the data in the QAPMJOBS file (or the QAPMJOBL file in V4R4 and beyond).

The following query will yield the information you need:

**Select SUM(JBCPU)/(SUM(INTSEC) *1000) as CPUPERCENT from QAPMJOBS where JBTYPE = "I".**

However, this will only show an average interactive utilization for the duration of a measurement interval (Smallest is 5 minutes, default is 15 minutes). Also, as will be described later in this section, the utilizations listed for job-type "I" are not necessarily all "interactive".

There are other means for determining interactive utilization more precisely. The easiest of these is the performance monitoring function of Management Central, which became available with V4R3. Management Central can provide:
- Graphical, real-time monitoring of interactive CPU utilization
- Creation of an alert threshold when an alert feature is turned on and the graph is highlighted
- Creation of an reverse threshold below which the highlights are turned off
- Multiple methods of handling the alert, from a console message to the execution of a command to the forwarding of the alert to another system.

By taking the ratio of the Interactive CPW rating and the Processor CPW rating for a system, one can determine at what CPU percentage the threshold is reached (This ratio works for the 7xx models and the current model 170 systems. For earlier models, refer to other sections of this document to determine what fraction of the Interactive CPW rating to use.) Depending on the workload, an alert can be set at some percentage of this level to send a warning that it may be time to redistribute the workload or to consider upgrading the interactive feature.

Another method is to combine the information provided by the WRKSYSACT command and the performance monitor. The WRKSYSACT command will give a fairly accurate accounting of work being done by each task in the system for intervals of 5 seconds or greater (A larger value is recommended for balancing the impact of the command with the work on the system, although it has the advantage over the performance monitor in that it only looks at active jobs, so it does not need to page in information about all jobs). The performance monitor database can determine which jobs are listed as interactive (In V4R3, use JBTYPE = "I" in the QAPMJOBS file. In V4R4, a more accurate determination can be made by examining JBSTSF = 1 in the QAPMJOBOS file. A join query between the file generated by WRKSYSACT (QAITMON) and the QAPMJxx file can give a fairly good picture of what the interactive utilization was when the measurement was taken.

With V4R4, the new Performance Collection functions that are available can yield similar results without having to run both the monitor functions and WRKSYSACT. The collection services functions can break the data down into very small time-slices (15 seconds), so the QAPMJOBOS file can be queried directly.

Finally, the functions of PM400 can also show the same type of data that the Performance Monitor shows, with the advantage of maintaining a historical view, and the disadvantage of being only historical. However, signing up for the PM400 service can yield a benefit in determining the trends of how interactive capacities are used on the system and whether more capacity may be needed in the future.

***Is Interactive really Interactive?***

Earlier in this document, the types of jobs that are classified as interactive were listed. In general, these jobs all have the characteristic that they have a 5250 workstation communications path somewhere within the job. It may be a 5250 data stream that is translated into html, or sent to a PC for graphical display, but the work on the AS/400 is fundamentally the same as if it were communicating with a real 5250-type display. However, there are cases where jobs of type "I" may be charged with a significant amount of work that is not "interactive". Some examples follow:

- Job initialization: If a substantial amount of processing is done by an interactive job's initial program, prior to actually sending and receiving a display screen as a part of the job, that processing may not be included as a part of the interactive work on the system. However, this may be somewhat rare, since most interactive jobs will not have long-running initial programs.

- More common will be parallel activities that are done on behalf of an interactive job but are not done within the job. There are two database-related activities where this may be the case.

    1. If the QQRYDEGREE system value is adjusted to allow for parallelism or the CHGQRYA command is used to adjust it for a single job, queries may be run in service jobs which are not interactive in nature, and which do not affect the total interactive utilization of the system. However, the work done by these service jobs is charged back to the interactive job. In this case, the performance monitor and most other mechanisms will all show a higher sum of interactive CPU

utilization than actually occurs. The exception to this is the WRKSYSACT command, which will show both the current activity for the service jobs and the activity that they have "charged back" to the requesting jobs. Thus, in this situation it is possible for WRKSYSACT to show a lower system CPU utilization that the sum of the CPU consumption for all the jobs.

2. A similar effect can be found with index builds. If parallelism is enabled, index creation (CRTLF, Create Index, Open a file with MAINT(*REBUILD), or running a query that requires an index to be build) will be sent to service jobs that operate in non-interactive mode, but charge their work back to the job that requested the service. Again, the work does not count as "interactive", but the performance data will show the resource consumption as if they were.

There are two key ideas in the statements above. First, if the workload has a significant component that is related to queries, it will be possible to show an interactive utilization in the performance tools that is significantly higher than what would be assumed from the ratings of the Interactive Feature and the Processor Feature. Second, although it may make monitoring interactive utilization slightly more difficult, in the case where the workload has a significant query component, it may be beneficial to set the QQRYDEGREE system value to allow at least 2 processes, so that index builds and many queries can be run in non-interactive mode. Of course, if the nature of the query is such that it cannot be split into multiple tasks, the whole query is run inside the interactive job, regardless of how the system value is set.

***How close to the threshold can a system get without disrupting performance?***

The answer depends on the dynamics of the workload, the percentage of work that is in queries, and the projected growth rate. It also may depend on the number of processors and the overall capacity of the interactive feature installed. For example, a job that absorbs a substantial amount of interactive CPU on a uniprocessor may easily exceed the threshold, even though the "normal" work on the system is well under it. On the other hand, the same job on a 12-way can use at most 1/12th of the CPU, or 8.3%. a single, intense transaction may exceed the limit for a short duration on a small system without adverse affects, but on a larger system the chances of having multiple intense transactions may be greater.

With all these possibilities, how much of the Interactive feature can be used safely? A good starting point is to keep the average utilization below about 70% of the threshold value (Use double the threshold value for the servers and earlier Model 170 systems that use the 1/3 algorithm described earlier in this document.) If the measurement mechanism averages the utilization over a 15 minute or longer period, or if the workload has a lot of peaks and valleys, it might be worthwhile to choose a target that is lower than 70%. If the measurement mechanism is closer to real-time, such as with Management Central, and if the workload is relatively constant, it may be possible to safely go above this mark. Also, with large interactive features on fairly large processors, it may be possible to safely go to a higher point, because the introduction of workload dynamics will have a smaller effect on more powerful systems.

As with any capacity-related feature, the best answer will be to regularly monitor the activity on the system and watch for trends that may require an upgrade in the future. If the workload averages 60% of the interactive feature with almost no overhead, but when observed at 65% of the feature capacity it shows some limited amount of overhead, that is a clear indication that a feature upgrade may be required. This will be confirmed as the workload grows to a higher value, but the proof point will be in having the historical data to show the trend of the workload.

## 2.11 Migration from Traditional Models

This section describes a suggested methodology to determine which server model is appropriate to contain the interactive workload of a traditional model when a migration of a workload is occurring.
It is assumed that the server model will have both interactive and client/server workloads.

To get the same performance and response time, from a CPU perspective, the interactive CPU utilization of the current traditional model must be known. Traditional CPU utilization can be determined in a number of ways. One way is to sum up the CPU utilization for interactive jobs shown on the Work with Active Jobs (WRKACTJOB) command.

```
**************************************************************************
                              Work with Active Jobs
                                                              10/22/97
CPU%: 33.0      Elapsed time: 00:00:00      Active jobs: 152

Type options, press Enter.

2=Change 3=Hold 4=End 5=Work with 6=Release 7=Display message
8=Work with spooled files 13=Disconnect ...

 Opt  Subsystem/Job     User       Type   CPU %  Function              Status
      BATCH             QSYS       SBS       0                       DEQW
      QCMN              QSYS       SBS       0                       DEQW
      QCTL              QSYS       SBS       0                       DEQW
        QSYSSCD         QPGMR      BCH       0  PGM-QEZSCNEP         EVTW
      QINTER            QSYS       SBS       0                       DEQW
        DSP05           TESTER     INT     0.2  PGM-BUPMENUNE        DSPW
        QPADEV0021      TEST01     INT     0.7  CMD-WRKACTJOB        RUN
      QSERVER           QSYS       SBS       0                       DEQW
      QPWFSERVSD        QUSER      BCH       0                       SELW
      QPWFSERVS0        QUSER      PJ        0                       DEQW
**************************************************************************
```

(Calculate the average of the CPU utilization for all job types "INT" for the desired time interval for interactive CPU utilization - "P" in the formula shown below.)

Another method is to run the Performance Monitor (STRPFRMON, ENDPFRMON) during selected time periods and review the first page of the Performance Tools/400 licensed program Component Report. The following is an example of this section of the report:

**********

Component Report
Component Interval Activity
Data collected 190396 at 1030

Member . . . : Q960791030  Model/Serial . : 310-2043/10-0751D Main St...
Library. . : PFR  System name. . : TEST01  Version/Re..

| ITV End | Tns/hr | Rsp/Tns | CPU % Total | CPU% Inter | CPU % Batch | Disk I/O per sec Sync | Disk I/O per sec Async |
|---|---|---|---|---|---|---|---|
| 10:36 | 6,164 | 0.8 | 85.2 | 32.2 | 46.3 | 102.9 | 39 |
| 10:41 | 7,404 | 0.9 | 91.3 | 45.2 | 39.5 | 103.3 | 33.9 |
| 10:46 | 5,466 | 0.7 | 97.6 | 38.8 | 51 | 96.6 | 33.2 |
| 10:51 | 5,622 | 1.2 | 97.9 | 35.6 | 57.4 | 86.6 | 49 |
| 10:56 | 4,527 | 0.8 | 97.9 | 16.5 | 77.4 | 64.2 | 40.7 |
| : | | | | | | | |
| 11:51 | 5,068 | 1.8 | 99.9 | 74.2 | 25.7 | 56.5 | 19.9 |
| 11:56 | 5,991 | 2.4 | 99.9 | 46.8 | 45.5 | 65.5 | 32.6 |

Itv End------Interval end time (hour and minute)
Tns/hr-------Number of interactive transactions per hour
Rsp/Tns-----Average interactive transaction response time

**********

(Calculate the average of the CPU utilization under the "Inter" heading for the desired time interval for interactive CPU utilization - "P" in the formula shown below.)

It is possible to have interactive jobs that do not show up with type "INT" or in the Performance Monitor Component Report. An example is a job that is submitted as a batch job that acquires a work station. These jobs should be included in the interactive CPU utilization count.

Most systems have peak workload environments. Care must be taken to insure that peaks can be contained in server model environments. **Some environments could have peak workloads that exceed the interactive capacity of a server model or could cause unacceptable response times and throughput.**

In the following equations, let the interactive CPU utilization of the existing traditional system be represented by percent P. A server model that should then produce the same response time and throughput would have a CPW of:

**Server Interactive CPW** = 3 * P * Traditional CPW

or for Custom Models use:

**Server Interactive CPW** = 1.0 * P * Traditional CPW (when P < 85%)

or

**Server interactive CPW** = 1.5 * P * Traditional CPW (when P >= 85%)

Use the 1.5 factor to ensure the custom server is sized less than 85% CPU utilization.

These equations provide the server interactive CPU cycles required to keep the interactive utilization at or below the knee of the curve, with the current interactive workload. The equations given at the end of the Server and Custom Server Model Behavior section can be used to determine the effective interactive

utilization above the knee of the curve. The interactive workload below the knee of the curve represents one third of the total possible interactive workload, for non-custom models. The equation shown in this section will migrate a traditional system to a server system and keep the interactive workload at or below the knee of the curve, that is, using less than two thirds of the total possible interactive workload. In some environments these equations will be too conservative. A value of 1.2, rather than 1.5 would be less conservative. The equations presented in the **Interactive Utilization** section should be used by those customers who understand how server models work above the knee of the curve and the ramifications of the V4R1 enhancement.

These equations are for migration of "existing workload" situations only. Installation workload projections for "initial installation" of new custom servers are generally sized by the business partner for 50 - 60% CPW workloads and no "formula increase" would be needed.

For example, assume a model 510-2143 with a single V3R6 CPW rating of 66.7 and assume the Performance Tools/400 report lists interactive work CPU utilization as 21%. Using the previous formula, the server model must have an interactive CPW rating of at least 42 to maintain the same performance as the 510-2143.

**Server interactive CPW = 3 * P * Traditional CPW**
**= 3 * .21 * 66.7**
**= 42**

A server model with an interactive CPW rating of at least 42 could approximate the same interactive work of the 510-2143, and still leave system capacity available for client/server activity. An S20-2165 is the first AS/400e series with an acceptable CPW rating (49.7).

Note that interactive and client/server CPWs are not additive. Interactive workloads which exceed (even briefly) the knee of the curve will consume a disproportionate share of the processing power and may result in insufficient system capacity for client/server activity and/or a significant increase in interactive response times.

---

## 2.12 Migration from Server Models

The section describes a recommended methodology for migrating from a server to a traditional model.

First determine the interactive CPU utilization for the server model. The second step is to determine the batch (client/server) CPU utilization for the server model. The previous section ("Migration from Traditional Models") describes how the Work with Active Jobs (WRKACTJOB) command or the Performance Monitor (STRPFRMON, ENDPFRMON) may be used to gather this information. The last step is to get the Maximum Client/Server CPW rating for the server.

Now in the following equations, let:

**I** = Interactive CPU Utilization
**B** = Batch CPU Utilization

**CPWcs** = Maximum Client/Server CPW rating

A traditional model that should produce the same response time and throughput would have a CPW of:
**Traditional CPW = CPWcs * (I + B) / .70**

Note: In the above formula the division by 70 percent (.70) is done as a guideline to keep the system's CPU utilization at 70 percent, or less.

For example, assume a model 170-2160 with a V4R2 Maximum Client/Server CPW rating of 114, and assume the Performance Tools/400 report lists interactive work CPU utilization as 10% and batch CPU utilization at 50%. Using the previous formula, the traditional model should have a CPW rating of at least 97.7 to maintain the same performance as the 170-2160, this corresponds to an AS/400e 620-2180 system.

**Traditional CPW = CPWcs * (I + B) / .70**
**= 114 * (.10 + .50) / .70**
**= 97.7**

This formula should ensure that this system will give similar performance, however; each situation is unique and should be evaluated with an understanding of what the performance goals are. For example, if longer batch execution times are acceptable then a system with a lower CPW rating may be sufficient.

---

## 2.13 Dedicated Server for Domino Performance Behavior

Five new DSD models have been announced with V5R1. These include the iSeries Model 270 with a 1-way and a 2-way feature, and the iSeries Model 820 with 1-way, 2-way, and 4-way features. In addition, OS/400 V5R1 is enhanced to bolster DSD server capacity for robust Domino applications that require Java Servlet and WebSphere Application Server integration. The new behavior which supports Domino-complementary workloads on the DSD is available after September 28, 2001 with the refreshed version of OS/400 V5R1. This enhanced behavior is applicable to all DSD models including the model 170 and previous 270 and 820 models. Additional information on Lotus Domino for AS/400 can be found in Chapter 11, “Domino for iSeries”.

For information on the performance behavior of DSD models for releases prior to V5R1, please refer the to V4R5 version of this document.

Please refer to Appendix D for performance specifications for the new DSD models, including the number of Mail and Calendaring Users (MCU) supported.

### 2.13.1 V5R1 DSD Performance Behavior

This section describes the performance behavior for all DSD models for the refreshed version of OS/400 V5R1 that is available after September 28, 2001.

A white paper, Enhanced V5R1 Processing Capability for the iSeries Dedicated Server for Domino, provides additional information on DSD behavior and can be accessed at:
**http://www.ibm.com/eserver/iseries/domino/pdf/dsdjavav5r1.pdf** .

**Domino-Complementary Processing**

Prior to V5R1, processing that did not spend the majority of its time in Domino code was considered non-Domino processing and was limited to approximately 10-15% of the system capacity. With V5R1, many applications that would previously have been treated as non-Domino may now be considered as Domino-complementary when they are used in conjunction with Domino. Domino-complementary processing is treated the same as Domino processing, provided it also meets the criteria that the DB2 processing is less than 15% CPU utilization as described below. This behavioral change has been made to support the evolving complexity of Domino applications which frequently require integration with function such as Java Servlets and WebSphere Application Server. The DSD models will continue to have a zero interactive CPW rating which allows sufficient capacity for systems management processing. Please see the section below on Interactive Processing.

In other words, non-Domino workloads are considered complementary when used simultaneously with Domino, provided they meet the DB2 processing criteria. With V5R1, the amount of DB2 processing on a DSD must be less than 15% CPU. The DB2 utilization is tracked on a system-wide basis and all applications on the DSD cumulatively should not exceed 15% CPU utilization. Should the 15% DB2 processing level be reached, the jobs and/or threads that are currently accessing DB2 resources may experience increased response times. Other processing will not be impacted.

Several techniques can used to determine and monitor the amount of DB2 processing on DSD (and non-DSD) iSeries servers for V4R5 and V5R1.

- Work with System Status (WRKSYSSTS) command, via the ***% DB capability*** statistic
- Work with System Activity (WRKSYSACT) command which is part of the IBM Performance Tools for AS/400, via the ***Overall DB CPU util*** statistic
- Management Central - by starting a monitor to collect the ***CPU Utilization (Database Capability)*** metric
- Workload section in the System Report which can be generated using the IBM Performance Tools for AS/400, via the ***Total CPU Utilization (Database Capability)*** statistic

**V5R1 Non-Domino Processing**

Since all non-interactive processing is considered Domino-complementary when used simultaneously with Domino, provided it meets the DB2 criteria, non-Domino processing with V5R1 refers to the processing that is present on the system when there is no Domino processing present. (Interactive processing is a special case and is described in a separate section below). When there is no Domino processing present, all processing, including DB2 access, should be less than 10-15% of the system capacity. When the non-Domino processing capacity is reached, users may experience increased response times. In addition, CFINT processing may be present as the system attempts to manage the non-Domino processing to the available capacity. The announced "Processor CPW" for the DSD models refers to the amount of non-Domino processing that is supported .

Non-Domino processing on the 270 and 820 DSD models can be tracked using the Management Central function of Operations Navigator. Starting with V4R5, Management Central provides a special metric called "secondary utilization" which shows the amount of non-Domino processing. Even when Domino processing is present, the secondary utilization metric will include the Domino-complementary processing. And , as discussed above, the Domino-complementary processing running in conjunction with Domino will not be limited unless it exceeds the DB2 criteria.

**Interactive Processing**

Similar to previous DSD performance behavior for interactive processing, the Interactive CPW rating of 0 allows for system administrative functions to be performed by a single interactive user. In practice, a single interactive user will be able to perform necessary administrative functions without constraint. If multiple interactive users are simultaneously active on the DSD, the Interactive CPW capacity will likely be exceeded and the response times of those users may significantly lengthen. Even though the Interactive CPW capacity may be temporarily exceeded and the interactive users experience increased response times, other processing on the system will not be impacted. Interactive processing on the 270 and 820 DSD models can be tracked using the Management Central function of Operations Navigator.

**Logical Partitioning on a Dedicated Server**

With V5R1, iSeries logical partitioning is supported on both the Model 270 and Model 820. Just to be clear, iSeries logical partitioning is different from running multiple Domino partitions (servers). It is **not** necessary to use iSeries logical partitioning in order to be able to run multiple Domino servers on an iSeries system. iSeries logical partitioning lets you run multiple independent servers, each with its own processor, memory, and disk resources within a single symmetric multiprocessing iSeries. It also provides special capabilities such as having multiple versions of OS/400, multiple versions of Domino, different system names, languages, and time zone settings. For additional information on logical partitioning on the iSeries please refer to *Chapter 18. Logical Partitioning (LPAR)* and **http://www.ibm.com/eserver/iseries/lpar**

When you use logical partitioning with a Dedicated Server, the DSD CPU processing guidelines are pro-rated for each logical partition based on how you divide up the CPU capability. For example, suppose you use iSeries logical partitioning to create two logical partitions, and specify that each logical partition should receive 50% of the CPU resource. From a DSD perspective, each logical partition runs independently from the other, so you will need to have Domino-based processing in each logical partition in order for non-Domino work to be treated as complementary processing. Other DSD processing requirements such as the 15% DB2 processing guidelines and the 15% non-Domino processing guideline will be divided between the logical partitions based on how the CPU was allocated to the logical partitions. In our example above with 50% of the CPU in each logical partition, the DB2 database guideline will be 7.5% CPU for each logical partition. Keep in mind that WRKSYSSTS and other tools show utilizations only for the logical partition they are running in; so in our example of a partition that has been allocated 50% of the processor resource, a 7.5% system-wide load will be shown as 15% within that logical partition. The non-Domino processing guideline would be divided in a similar manner as the DB2 database guideline.

**Running Linux on a Dedicated Server**

As with other iSeries servers, to run Linux on a DSD it is necessary to use logical partitioning. Because Linux is it's own unique operating environment and is not part of OS/400, Linux needs to have its own logical partition of system resources, separate from OS/400. The iSeries Hypervisor allows each partition to operate independently. When using logical partitioning on iSeries, the first logical partition, the primary partition, must be configured to run OS/400. For more information on running Linux on iSeries, please refer to *Chapter 13. iSeries Linux Performance* and **http://www.ibm.com/eserver/iseries/linux** .

Running Linux in a DSD logical partition will exhibit different performance characteristics than running OS/400 in a DSD logical partition. As described in the section above, when running OS/400 in a DSD logical partition, the DSD capacities such as the 15% DB2 processing guideline and the 15% non-Domino processing guidelines are divided proportionately between the logical partitions based on how the processor resources were allocated to the logical partitions. However, for Linux logical partitions, the DSD guidelines are relaxed, and the Linux logical partition is able to use all of the resources allocated to it outside the normal guidelines for DSD processing. This means that it is not necessary to have Domino

processing present in the Linux logical partition, and all resources allocated to the Linux logical partition can essentially be used as though it were complementary processing. It is not necessary to proportionally increase the amount of Domino processing in the OS/400 logical partition to account for the fact that Domino processing is not present in the Linux logical partition .

By providing support for running Linux logical partitions on the Dedicated Server, it allows customers to run Linux-based applications, such as internet fire walls, to further enhance their Domino processing environment on iSeries. At the time of this publication, there is not a version of Domino that is supported for Linux logical partitions on iSeries.

# Chapter 3. Batch Performance

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

**Batch Workload Description**

The Batch Commercial Mix is a synthetic batch workload designed to represent multiple types of batch processing often associated with commercial data processing. The different variations allow testing of sequential vs random file access, changing the read to write ratio, generating "hot spots" in the data and running with expert cache on or off. It can also represent some jobs that run concurrently with interactive work where the work is submitted to batch because of a requirement for a large amount of disk I/O.

## 3.1 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 scales inversely with Relative Performance Rating (CPWs). This assumes that the number synchronous disk I/Os are only a small factor.

- 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.

- It is recommended that capacity planning for batch be done with tools that are available for AS/400. For example, BEST/1 for OS/400 (part of the Licensed Program Product, AS/400 Performance Tools) can be used for modeling batch growth and throughput. BATCH400 (an IBM internal tool) can be used for estimating batch run-time.

## 3.2 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

Additional information on DASD devices in a batch environment can be found in Chapter 14, “DASD Performance”.

## 3.3 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 did not have a significant effect on the Commercial Mix batch workload. 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. In the Commercial Mix benchmark, the CPU was the limiting factor.

  However, specific batch environments that are DASD I/O intensive, and process data sequentially may realize significant performance gains by taking advantage of larger memory sizes available on the RISC models, particularly at the high-end. Even though in general applications require more main storage on the RISC models, batch applications that process data sequentially may only require slightly more main storage on RISC. Therefore, with larger memory sizes in conjunction with using Expert Cache, these applications may achieve significant performance gains by decreasing the number of DASD I/O operations.

- **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.

---

## 3.4 V4R4 Comments

We observed an increase in the CPU requirements for traditional (RPG and COBOL end-of-day processing) batch workloads of 5-8%. Except for environments where the system is nearing the need for an upgrade or environments where a particular job must finish prior to other jobs starting, we do not expect this to have a major effect on the overall batch window.

# Chapter 4. DB2 UDB for iSeries Performance

This chapter contains performance information on items that are important to achieving a good overall level of performance for DB2 UDB for iSeries environments. The information presented here concentrates on performance for applications run locally on DB2 UDB for iSeries, although much of the information can also be used to ensure better levels of performance for applications using remote access to an iSeries.

The first section in this chapter provides information on what has changed in DB2 UDB for iSeries in V5R1. The second section concentrates on DB2 enhancements made in Version 4 (V4R1 through V4R5). The last section contains articles that were written in past (Version 3) releases about DB2 for AS/400 performance characteristics.

## 4.1 V5R1 Enhancements for DB2 UDB for iSeries

In V5R1, enhancements were made that will help improve performance for DB2 UDB for iSeries, primarily in the area of SQL and query performance. This section will describe what these changes are and what environments they will affect. If you want to learn more about performance of DB2 UDB for AS/400, refer to the items described later in this section under the title "Sources of Additional Information".

**Performance Improvements in V5R1**

Although performance measurement data is not available for these enhancements, these changes will in many cases result in noticeable improvements for environments that are able to take advantage of them.

1. FETCH FIRST N ROWS ONLY clause

   Starting in V5R1, users will be able to specify the FETCH FIRST N ROWS ONLY clause on their SELECT statements. This feature allows users to limit the size of the result set of a query to a specified number of rows. Use of this clause can improve performance for queries that have potentially large results when only a limited number of rows is of interest to the end user or application. For example, a user might be interested in viewing only information on the ten highest paid employees in an organization. In this case, the user could just issue a SELECT statement with a FETCH FIRST 10 ROWS ONLY specified. For clients issuing SELECT statements to a server, use of this clause can also help reduce the amount of data sent back to the client, which can help improve response time as well.

   Note that for queries involving subselects, the FETCH FIRST N ROWS ONLY is only allowed on the outermost SELECT and not on the inner subselects. In addition, the use of this clause will not affect how the query optimizer views a given query. For example, the optimizer will not favor an index over a query sort based on the number of records specified in this clause.

2. Improvements for INSERT with subselect

   In V5R1, changes have been made to increase the size of the IO buffers used for both the from and to files for an INSERT with subselect operation. For cases where large amounts of data are being read

and inserted, this enhancement can result in significant reductions in run time.

3. Join optimization improvements

   The query optimizer may now do partial join reordering for queries that involved inner joins along with left outer joins and/or exception joins. Prior to this enhancement, the tables in the inner join were forced to be joined in the order they were specified because of the existence of the left outer join or the exception join in the query. With this change, the optimizer may now choose to reorder the tables in the inner join to achieve optimal performance levels. Queries that were previously locked into the specified join order may now realize significant performance improvements if a better join order is available for the optimizer to consider.

   Also in V5R1, the OS/400 licensed internal code was enhanced to support joins of up to 256 files. Prior to V5R1, the query optimizer allowed joins of more than 32 files, but this was done by optimizing the first group of 32 joined files as listed in the query and placing those results in a temporary file, then processing subsequent groups of 32 the same way and joining the temporary files at the end of the query. In V5R1, the new support enables the optimizer to look at all the files and put the optimal 32 files at the front of the join, and then run the query as a single large join involving all the files. In many cases, the improvement in performance for these types of joins can be significant.

4. Increased use of indexes for some SQL expression

   Prior to V5R1, the query optimizer was able to use indexes for simple SQL expressions that involve the use of functions such as CAST, CHAR, VARCHAR, GRAPHIC or VARGRAPHIC for literal values, but not for host variables or parameter markers. In V5R1, changes have been made that allow the optimizer to consider the use of an index for these expressions no matter what type of value is specified to be converted or translated. Since many OS/400 SQL interfaces substitute parameter markers in place of literals for performance reasons, this enhancement can be important for applications that use these SQL functions via these interfaces.

5. Improvements for queries using LIKE conditions

   Prior to V5R1, there were certain patterns used in LIKE conditions that always resulted in non-reusable ODPs for the query containing them. In V5R1, these restrictions have been removed such that any LIKE pattern will no longer cause use of non-reusable ODPs. In addition, other enhancements were made for key range selection with LIKE predicates (for example, indexes can now be used for all LIKE conditions containing double-byte characters). These enhancements can result in significantly better performance for queries with LIKE predicates.

6. Caching of query index estimates

   In many cases, a significant portion of the cost of doing query optimization is due to the optimizer needing to do key range estimates on indexes to determine which indexes are useful for implementing a query. Prior to V5R1, these key range estimates were repeated each time a query was optimized, sometimes resulting in excessive amounts of CPU and IO being used. In V5R1, enhancements have been made to cache the results of key range estimates as part of the index object itself. Each new key range estimate request for an index will search this cache first to see if a usable estimate already exists, thus potentially avoiding the more time consuming index probes. This enhancement can result in

noticably better performance for applications that repeatedly optimize the same query or similar queries that use the same key ranges.

7. Faster handle allocation in CLI

   In V5R1, the amount of CPU resource used for allocating and deallocating statement handles in CLI applications has been significantly reduced. The amount of improvement from this change will depend on how often statement handles are allocated in the end user applications. For applications where this type of operation is done frequently, the improvement in CPU and response time can be significant.

8. Journalling performance improvments

   V5R1 contains enhancements which make a rollback of tentative database changes made under commitment control nearly 8 times faster in some environments.

   V5R1 provides support for a new API, Call QJOCHRVC 1000000, which can be employed to reduce background journal disk traffic. This reduced traffic can be especially helpful when executing large batch jobs.

   V5R1 supports the Aggressive Journal Caching PRPQ, 5799-BJC, which can substantially reduce the number of synchronous disk writes your job must wait for. By reducing the number of disk writes to the journal receiver, both your batch and interactive jobs can complete in less time. Measured results have varied from a 20% reduction in elapsed time for some batch jobs up to a 75% reduction in some cases.

   V5R1 provides support for a new Journal option: MINENTDTA , which enables the Journal_Minimal_Data option. Use of this option can reduce the total disk traffic as well as the number of bytes sent to the journal. It's especially helpful in batch jobs which tend to update existing database records rather than add new records. This option journals only the changed bytes within the updated record thereby reducing the size of each journal entry. This will be especially attractive for applications which tend to update only a few status or quantity fields within most database records while leaving the rest of the record's fields unchanged. Since the journal receivers fill less rapidly, your applications will be slowed down less often to swap journal receivers.

## Sources of Additional Information

1. For further information on other DB2 UDB for iSeries enhancements, see the Internet page at the following url:

   http://www.iSeries.ibm.com/db2

   This page contains references and urls that can be followed to obtain information such as recent announcements, articles and white papers, technical information and tips, and further enhancements and performance improvements for DB2 UDB for iSeries.

   In addition, information on educational offerings and classes that will help with developing SQL and query performance tuning skills are available at the following urls:

http://www.iSeries.ibm.com/db2/db2educ_m.htm
http://www-3.ibm.com/servlet/com.ibm.ls.lsow.servlets.CourseDescriptionServlet?coursecode=S6140

2. Some of the above enhancements as well as other potential query performance improvements may be available in what are called database fixpack PTF packages for previous release levels. To find out more, refer to the item on these fixpacks at the following url:

   http://www.iSeries.ibm.com/db2/db2tch_m.htm

   In addition to fixpack information, this url points to other items such as technical overview articles, publications and redbooks and other enhancements, both functional and performance related.

3. The iSeries Teraplex Center plays a significant role in verifying the benefits of new technologies for data warehousing operations. In many cases their testing applies to other general applications of these technologies as well. For a variety of recent test results and additional information on many of the new see the Teraplex Center's Internet home page at url:

   http://www.iSeries.ibm.com/developer/bi/teraplex

4. The iSeries Systems Performance area maintains an IBM Internet home page. For a variety of white papers and additional information on many of the technologies described in this chapter, see the performance section at url:

   http://www.ibm.com/servers/eserver/iseries/perfmgmt/

5. For additional information and guidance about database and query performance tuning, refer to the DB2 UDB for iSeries Database Performance and Query Optimization manual. HTML and PDF versions of this and other DB2 UDB for iSeries manuals can be found in the V5R1 iSeries Information Center at url:

   http://submit.boulder.ibm.com/pubs/html/as400/bld/v5r1/ic2924/index.htm

---

## 4.2 Version 4 Enhancements for DB2 UDB for AS/400

### Performance Improvements in V4R5

Although performance measurement data is not available for these enhancements, these changes will in many cases result in noticeable improvements for environments that are able to take advantage of them.

1. New options added to INI file interface

   Two new options, OPEN_CURSOR_THRESHOLD and OPEN_CURSOR_CLOSE_COUNT, been added to the query INI file in V4R5. These options can be used by customers to set limits on how many reusable SQL cursors are allowed in their jobs and how many cursors should be closed when this limit

is reached. For environments with potentially large numbers of open cursors, proper use of these options can significantly reduce the amount of overhead and resources used by SQL to internally handle and maintain these cursors. This reduction in overhead can result in noticeable overall performance gains for these environments.

For more information on these and other options available via the INI file interface, refer to the DB2 UDB for AS/400 SQL Programming manual (SC41-5611).

2. Improved tools for query performance analysis

   In V4R5, a new tool named Visual Explain was added to the AS/400 Operations Navigator to help make AS/400 query performance analysis much easier. This tool allows users to not only see what queries are using the most time and resource in their environments, but also to quickly analyze how individual queries were optimized and what can be done to tune these queries for better performance. Users now have the ability to visually see how an individual query was optimized and also to view other more detailed information on how the query was executed. Use of this tool to determine what queries are costly and how to improve performance for these queries can result in large gains in performance for many SQL applications and environments. In addition, the output provided by the AS/400 database monitor is significantly enhanced in V4R5 to provide much more detailed query performance and tuning information, both for use by the Visual Explain tool and by users who previously were using database monitor data to do their own query analysis.

   Note that support for visually explaining certain types of queries (subqueries, inserts with subselects, DB2 Multisystem queries and hash join queries) is not currently available via the Visual Explain tool.

3. Improvements in dynamic SQL

   Changes have been made in V4R5 to allow parameter marker conversion to occur in more cases for dynamic SQL statements, and also to help reduce internal maintenance costs for some dynamic SQL environments. For some applications, these changes may result in a significant drop in the number of full opens and closes, as well as improvements in prepare and/or describe time. For applications or environments affected by these enhancements, the amount of improvement in CPU, IO and run time can be large.

4. Larger index page sizes for SQL tables

   In V4R5, changes were made to enable DB2 UDB for AS/400 to create indexes (also known as logical files or access paths) with a larger logical page size (64K). Since this larger page size results in better database paging, this means that the query optimizer will often choose these indexes for running a query. For queries that use indexes with the larger page sizes, the improvement in run time can be dramatic compared to the same query using the previously available indexes where page sizes were restricted to 4K or 8K.

   This new larger page size will only be used on indexes created over SQL tables and SQL-created indexes. The 64K logical page size will be used automatically for any index that is created over an SQL-created table - this includes both SQL-created indexes and indexes created for constraints. For more information on creating these indexes, refer to the following url:

   http://www.as400.ibm.com/db2/ixszptf.htm

Note that as indexes are rebuilt with 64K page sizes, some of the potential query improvement may be offset for indexes whose keys are frequently updated via updates or inserts to the SQL table. For these indexes, additional seize contention may occur, which can have negative effects on response time. There may also be increased amounts of journal data being generated if the indexes are journaled either via SMAPP or via explicit user journaling. This additional data may result in journal receivers filling up more quickly than in the past.

5. Reusable ODPs allowed for more types of queries

   Prior to V4R5, queries that required the optimizer to use a temporary file resulted in that query being run with a non-reusable ODP, which meant a full open and close for each execution of the query. Examples include queries that did a group by on columns from more than one table, or those that used views whose contents needed to be materialized into a temporary file in order to run the query. In V4R5, these types of queries can now be run with a reusable ODP. For some applications, this improvement can result in large performance improvements.

   Also prior to V4R5, there were certain patterns used in LIKE conditions that always resulted in non-reusable ODPs for the query containing them. In V4R5, these restrictions have been removed such that any LIKE pattern will no longer cause use of non-reusable ODPs. In addition, other enhancements were made for key range selection with LIKE predicates. Both enhancements can result in significantly better performance for queries with LIKE predicates.

6. Improved performance for many join queries

   In V4R5, enhancements have been made to allow multi-key row positioning selection (also know as MKF) to be used in all types of query joins. Basically, this allows the query optimizer to more consistently find and apply selection against join criteria used within a given query, and also to more consistently choose an exact set of key fields to perform the join and selection with. Overall, these changes will result in performance improvements for many join queries, although the amount of improvement will vary depending on how much help additional selection provides for each query.

7. Improvements in EVI maintenance

   Although encoded vector indexes (EVIs) do in many cases provide large boosts in query performance, one possible disadvantage of having them present over a given SQL table is the potential slowdown in performance for INSERTs, DELETEs and UPDATEs for that table when these operations result in excessive maintenance costs for the EVIs. In V4R5, changes have been made to internal EVI maintenance routines to help improve performance for these types of change operations when EVIs exist over the SQL table. Although the amount of improvement will vary, the fact that the maintenance costs have been reduced may now make EVIs a more viable option for some SQL tables that previously could not use EVIs because of the overhead of maintaining them.

8. Maximum journal receiver size increased

   In V4R5, the maximum size allowed for a journal receiver has been increased from 2 gigabytes to 1 terabyte. Prior to V4R5, customers with large volumes of journal entries who chose to employ user-managed journal receivers for their environments had to frequently issue a CHGJRN command to attach a new journal receiver when the current journal receiver filled up. These operations could be

disruptive because the underlying database code had to ensure that all changed pages for all the journaled objects were written to disk before the CHGJRN could complete, which meant that no new database records could be added or updated until this was done. Given a maximum of 2 GB in size, these disruptive operations could occur frequently for these users. Now that the maximum size of the journal receiver is 1 TB, this can mean less CHGJRN operations being required and significantly less disruption to those customers who currently employ user-managed journal receivers with large volumes of journal activity.

## Performance Improvements in V4R4

In V4R4, DB2 for AS/400 was renamed to DB2 Universal Database for AS/400 (DB2 UDB for AS/400). Following is a description of the changes that were made to help improve query performance in V4R4. Enhancements were made in this release to help improve performance for both business intelligence (BI) queries as well as for those queries likely to be found in everyday batch and OLTP applications. Although performance measurement data is not available for these enhancements, these changes will in many cases result in noticeable improvements for queries that are able to take advantage of them.

1. General improvement in query optimization and runtime system programs

   Many of the query optimization and runtime programs have been changed to use a new IBM internal programming language to take advantage of better compiler optimization. This has resulted in noticeable performance improvements in some areas, particularly in the amount of CPU used during query optimizations.

2. Reduction in memory used by queries

   Changes have been made to internal space allocation algorithms and to internal query structures that will help reduce the memory footprint for most queries. This will help reduce overall memory consumption, especially in environments running with large numbers of reusable ODPs.

3. Improved run times for some temporary sorts

   On the AS/400, queries using temporary sorts often show a relatively high cost at open time, even when the open is for a reusable ODP. A significant portion of this cost is due to initializations required for the sort, and in cases where the sort only involves a few records, the initialization cost accounted for a large portion of the open. In V4R4, temporary sorts involving a small number of rows will now use a different sort algorithm that will significantly reduce this cost at open time. For environments that involve a significant number of reusable ODPs with small temporary sorts, this enhancement could result in noticeable improvements in overall run time.

4. Increased use of hash join

   The use of a hash join will in many cases be more efficient than a nested loop join. However, prior to V4R4, hash joins were not allowed for queries running with commitment control levels of *CHG or *CS or for queries with subqueries where the entire query was implemented as a composite join. In V4R4, these restrictions have been removed, which may result in improved performance for those

queries which previously ran under these limitations but could have benefited from using a hash join.

5. Improved performance for MIN and MAX functions

   In V4R4, queries that involve use of the grouping functions MIN or MAX for a given column will in some cases realize a performance improvement provided that the grouped column is retrieved using an index. To find out more about what types of queries will benefit from this change, refer to the DB2 UDB for AS/400 SQL Programming manual (SC41-5611) for V4R4.

6. Improvements for partial outer and exception joins

   In V4R4, the optimizer can now implement partial outer (PO) joins and exception (EX) joins using using a join method known as the key positioning access method. This join method existed prior to V4R4 and was used in other join types, but were not allowed in PO or EX joins. Use of the key positioning access method in these join types and in queries with mixtures of these joins can in many cases result in significant performance improvements.

7. Better use of EVIs and dynamic bitmaps

   In V4R4, changes have been made to allow improved use of EVIs and dynamic bitmaps. For example, multiple EVIs built over both the fact and dimension tables can be used to generate a list of all possible values to be selected from the fact table. This list is then used to generate a bitmap that can be used in scanning the fact table, which can result in noticable performance gains. Performance of some join queries, in particular star schema joins, may also show significant improvements from these changes.

8. Improved internal handling of SQL opens and cursors

   Changes have been made in V4R4 to enable parameter marker conversion to occur in more cases, which will help reduce the number of full opens that occur in some dynamic or extended dynamic environments. Also, improvements have been made to the algorithms used to find an existing open cursor (reusable ODP) within a job. For jobs with a large number of reusable ODPs, this improvement can be noticeable.

9. Improved interface for debugging query performance

   A new INI file interface is available in V4R4 that provides users with the ability to dynamically modify or override the environment in which queries are executed. This support is shipped with the V4R4 base release. This interface also allows Rochester lab developers the ability to control and debug queries for performance without having to install additional tools on the system. For more information on this interface and how to use it, refer to the DB2 UDB for AS/400 SQL Programming manual (SC41-5611) for V4R4.

## Encoded Vector Indices (EVIs)

In V4R3 a new type of permanent index, the Encoded Vector Index (EVI), can be created through SQL. EVIs cannot be used to order records, but in many cases, they can improve query preformance. An EVI has several advantages over a traditional binary radix tree index.

- The query optimizer can scan EVIs and automatically build dynamic (on-the-fly) bitmaps much more quickly than traditional indexes. For more information on dynamic bitmaps, see the description in their section below.

- EVIs can be built much faster and are much smaller than traditional indexes. Smaller indexes require less DASD space and also less main storage when the query is run.

- EVIs automatically maintain exact statistics about the distribution of key values, whereas traditional indexes only maintain estimated statistics. These EVI statistics are not only more accurate, but also can be accessed more quickly by the query optimizer.

EVIs are used by the AS/400 query optimizer with dynamic bitmaps and are particularly useful for advanced query processing. EVIs will have the biggest impact on the complex query workloads found in business intelligence solutions and adhoc query environments. Such queries often involve selecting a limited number of rows based on the key value being among a set of specific values (eg a set of state names).

When an EVI is created and maintained, a symbol table records each distinct key value and also a corresponding unique binary value (the binary value will be 1, 2, or 4 bytes long, depending on the number of distinct key values) that is used in the main part of the EVI, the vector (array). The subscript of each vector (array) element represents the relative record number of a database table row. The vector has an entry for each row. The entry in each element of the vector contains the unique binary value corresponding to the key value found in the database table row.

The following is an example of how to create an EVI with the SQL CREATE INDEX statement:

```
CREATE ENCODED VECTOR INDEX StateIx
ON CUSTOMERS (CustState)
```

**Parallel Data Loader**

The data loader is a new function in V4R3 (also available via PTF on V4R2 and V4R1) that makes loading AS/400 database tables from external data much simpler and faster. The data loader can import fixed-format, delimited, and byte-stream files. A new CL command, Copy From Import File (CPYFRMIMPF), is provided to simplify the process.

After installing and activating the DB2 for AS/400 SMP licensed feature on a multiprocessor AS/400, parallel processing increases the speed of the data loader by approximately ten times over non-parallel methods. With this feature active, DB2 for AS/400 is able to use multi-tasking, rather than just a single task, to load an import file.

The following PTFs are required to provide these functions on earlier releases:

- V4R1M0 PTFs: SF47138 and SF47177 for OS/400
- V4R2M0 PTFs: SF46911 and SF46976 for OS/400

These are available individually and may be available in a cumulative\pard PTF package.

## Parallel Index Maintenance

Parallel index maintenance, supported in V4R3, can be useful to those that have many logical files and indexes defined over a single database file. Every time that a row is inserted, changed or deleted into a database table, all of the indexes and logical files defined over that base table have to be maintained to reflect the latest data change. The parallel index maintenance enhancement allows DB2 for AS/400 to maintain multiple indexes in parallel instead of one at a time as done in previous releases. Note that DB2 for AS/400 will utilize parallel index maintenance only when blocked insert operations are performed on the base database table, there exist at least eight indexes over the table, and the DB2 for AS/400 SMP (Symmetric Multiprocessing) licensed feature is installed and activated. Parallel index maintenance thus allows DB2 for AS/400 to reduce the amount of time it takes to maintain indexes when you are adding lots of new rows to a database table. The data loader and copy file (CPYF) utilities also will benefit from this feature since they utilize blocked inserts.

## Dynamic Bitmaps

Dynamic bitmaps, introduced in V4R2, can improve the performance of certain query operations. A dynamic (on-the-fly) bitmap is a temporary binary structure built against a permanent index. The AS/400 database query optimizer automatically builds dynamic bitmaps when the optimizer determines that dynamic bitmaps will speed up response time. Use of dynamic bitmaps allows the system to perform skip-sequential DASD operations and reduces the need for full-table-scans, thereby reducing database I/O operations and speeding up completion of the affected queries. In addition, DB2 for AS/400's dynamic bitmap support allows the use of multiple indexes against any particular table in the query (previously limited to at most one index per table). These multiple bitmaps resulting from the use of more than one index are combined into a composite results bitmap using boolean logic.

When the optimizer builds a dynamic bitmap, it sets a bit for every index entry that meets the selection criteria. It can combine (AND, OR) multiple bitmaps into a composite results bitmap. The optimizer uses this final bitmap to retrieve only those records whose bits are set.

Note that single-column indexes can often maximize flexibility for the database administrator. Dynamic bitmaps allow the optimizer to use several of these single-column indexes at once and combine their dynamic bitmaps, using boolean logic, into one bitmap. In this way, ad hoc users of large databases may be able to realize large performance gains without significant impact to the system and with a minimal set of indexes.

## System-Wide SQL Statement Cache

The system-wide SQL statement cache, introduced in V4R2, can improve the performance of programs using dynamic SQL. The system automatically caches dynamic SQL statements. No user action is required to activate or administer this function.

When a dynamic SQL statement that has previously executed is later reexecuted, if the statement is still available in the cache, DB2 for AS/400 can retrieve (rather than construct) key information associated with the cached statement, and thereby reduce the processing resource and the time required to execute the statement again.

## Remote Journal Function

Introduced in V4R2, the remote journal function allows replication of journal entries from a local (source) AS/400 to a remote (target) AS/400 by establishing journals and journal receivers on the target system that are associated with specific journals and journal receivers on the source system. Some of the benefits of using remote journal include:

- Allows customers to replace current programming methods of capturing and transmitting journal entries between systems with more efficient system programming methods. This can result in lower CPU consumption and increased throughput on the source system.

- Can significantly reduce the amount of time and effort required by customers to reconcile their source and target databases after a system failure. If the synchronous delivery mode of remote journal is used (where journal entries are guaranteed to be deposited on the target system prior to control being returned to the user application), then there will be no journal entries lost. If asynchronous delivery mode is used, there may be some journal entries lost, but the number of entries lost will most likely be fewer than if customer programming methods were used due to the reduced system overhead of remote journal.

- Journal receiver save operations can be off-loaded from the source system to the target system, thus further reducing resource and consumption on the source system.

Hot backup, data replication and high availability applications are good examples of applications which can benefit from using remote journal. Customers who use related or similar software solutions from other vendors should contact those vendors for more information.

### Test Environment

As mentioned above, remote journal can be run in either synchronous delivery mode or asynchronous delivery mode. In synchronous mode, the source system must wait for a confirmation message from the target system that the journal entries have been received. In asynchronous mode, the source system runs without having to wait for remote journal to finish. In both sync and async modes, remote journal can be in one of two activation modes. If the source journal receivers contain entries that have not been replicated at the time remote journal is started, then remote journal will run in "catch-up" mode to transfer these entries to the remote system. Once remote journal catches up with the source system journal entries, it then runs in "continuous" mode.

Tests were done in Rochester to evaluate the following aspects of remote journal performance:

- Comparing elapsed times of running catch-up mode using TCP/IP, APPC and Opticonnect for OS/400

- Overall impact of running remote journal in continuous mode in an interactive transaction processing environment using TCP/IP, APPC and Opticonnect for OS/400

For all tests, the source system used was a model 530-2151 with 2.5 GB of memory and 73 GB of DASD (24 arms totalling 65 GB in the system ASP, 4 arms totalling 8 GB in a user ASP). The target system was a model 510-2143 with 1.0 GB of memory and 33 GB of DASD (16 arms totalling 25 GB in the system ASP, 4 arms totalling 8 GB in a separate user ASP). All DASD arms on both systems were unprotected

(RAID or mirroring was not used). Both systems were installed with V4R2, and on both systems, the journal receivers were located on the user ASP.

For both the TCP/IP and APPC tests, a 16Mbps token ring was used (model 2629 with 6149 cards on both systems). For Opticonnect, the bus model used was a 2682, with the source system using a 2685 card and the target system using a 2688 card.

Note that the results shown here are for specific environments and configurations running a controlled and repeatable series of tests. The actual results you obtain in your environment may vary from what is presented here, although the conclusions and recommendations made here will be applicable to most customer applications using remote journal.

**Remote Journal Catch-Up Mode**

For the catch-up mode tests, approximately 3.1 GB of journal entries was transferred to the remote system when remote journal was started. Memory pools were allocated to ensure this resource was not constrained. There was no other activity on either the source or target systems.

Using TCP/IP and APPC, this transfer took just over 26 minutes, while it took about 12 minutes using Opticonnect. These results are as expected given the much faster transfer rates of the Opticonnect bus versus the token ring connection. In all cases, the CPU utilization was low (15% in the Opticonnect case and less than 10% in the other measurements). No other system resources were constrained during these tests.

**Remote Journal Continuous Mode**

The base run for this test was done using an interactive transaction processing environment with 640 users running about 224,000 transactions per hour on simulated locally attached 5250 workstations. All files that were updated in this environment were journaled to a single local journal receiver on the source system's user ASP. Commitment control was not used. This environment produces about 3.2 million journal entries per hour with a CPU utilization of about 72% on the source system. Memory and DASD are not constrained in this environment. The response time for this environment is based on one transaction type that accounts for 45% of the total throughput and produces about 35 journal entries per transaction. In this environment, the average response time for this transaction was 0.15 seconds.

For the remote journal runs, all journal entries produced on the source system were replicated on the target system. Both asynchronous mode and synchronous mode were tested on each of the three different communications protocols. In the remote journal runs, the target journal receiver was located on the user ASP on the target system, and again memory and DASD were not constrained. There was no other activity on the target system.

When remote journal with asynchronous mode was added to the base environment, the results show that for all three protocols, the impact to response time was minimal (about 0.05 seconds) with a corresponding drop in throughput of less than 1%. The increase in CPU utilization on the source system ranged from 2-7%, and the token ring and bus utilizations were very reasonable (less than 15% in all cases). The target system in all three environments showed a CPU utilization of 2-3% with low levels of disk arm utilization.

When remote journal with synchronous mode was added to the base environment, the Opticonnect run showed the least impact to response time (increase of about 0.15 seconds) with a corresponding drop in

throughput of about 1.5%, while TCP/IP and and APPC showed an increase of about 0.3 seconds with a drop in throughput of about 3%. In all three environments, the CPU overhead per transaction from adding remote journal was 7-8% on the source system. The token ring utilization in the APPC and TCP runs was about 15-20%, and the bus utilization was also low in the Opticonnect run. The CPU overhead on the target system was about 5-7%, and DASD utilizations on the target system user ASP were again low as they were in the async mode runs.

**Conclusions and Recommendations**

- For most interactive environments, adding remote journal should not result in significant degradations to response time or resource utilization on either the source or target system. Although you can expect a more noticeable increase in response times when using synchronous mode, the tests above show that the amount of the increase should still be reasonable. In sync mode, the amount of increase will be relative to the number of journal entries produced by an average transaction.

  One other item to consider is that the interactive transactions described here spend over 70% of their time doing database activity, with 20% or so spent in application code. In many customer environments, the average transaction spends a much higher percentage of time in application code and other areas and a much lower percentage in database activity. Overall, this means that for many customer applications, the overall impact from adding remote journal may be less than it was in the tests described here.

  There are several factors to consider when deciding what sort of remote journal setup you will choose:

  1. Whether or not you can afford loss of some journal entries in the event of a system failure. Synchronous mode will guarantee no journal entries are lost, while asynchronous mode may lose some entries (although the number will be relatively low compared to the total number of journal entries being replicated).

  2. In general, Opticonnect offers noticeable advantages over APPC or TCP/IP alternatives via token ring, including less of an impact to response times (particularly when using sync mode) over a broader range of throughput and also higher overall capacity. For customers who already have Opticonnect installed and have additional capacity available, this would be a logical choice for implementing remote journal. However, users who do not currently have Opticonnect need to balance the added cost of this product versus what can be accomplished over a token ring using APPC or TCP/IP. For example, if the token ring has enough capacity for your level of remote journal and you either use async mode or can afford the response time increase with sync mode, then using APPC or TCP/IP with the token ring would also be a logical choice.

  3. Another item to consider for remote journal is DASD performance, particularly when using Opticonnect. Since Opticonnect allows a much higher transfer rate and capacity without bottlenecking the bus, the performance of the DASD arms where the journal receiver is located can become a problem in very heavy journaling environments, particularly if the arms are RAID protected. In heavy remote journal environments, Opticonnect can achieve transfer rates up to 1 GB per second when the journal receivers on both systems are located on unprotected DASD arms (no RAID or mirroring). If RAID protection is used in the same environment, the DASD arms become the bottleneck. If protection is required, using mirrored arms for the journal receiver DASD will provide significantly better performance than RAID protection. However, unprotected DASD arms will allow the highest overall transfer rates and best overall performance when using

Opticonnect for remote journal. Whatever implementation is used, DASD arm performance should still be monitored using the performance monitor and/or other tools such as the WRKDSKSTS command.

In most cases involving high transfer rates via TCP/IP or APPC, the communications line or IOP will tend to become the bottleneck prior to the journal receiver DASD arms becoming overutilized. However, it is still a good idea to monitor the performance of these arms to ensure the best overall results.

4. In catch-up mode, performance will be significantly better using Opticonnect than with TCP or APPC due to the higher transfer rates and overall higher capacity of the Opticonnect bus. This should also be considered when determining what medium to use in your overall remote journal implementation.

5. The measurements done here show the impact of adding remote journal to an application that is doing local journaling without commitment control. Similar results can be expected when adding remote journal to customer applications that use commitment control.

- If remote journal will be used during execution of a batch job, it is recommended that asynchronous mode be used. The reason for this is that in the event of a system failure, most customers choose to start the batch job over from the beginning instead of restarting the batch job partway through after recovering to that point using the remote journal. Because of this, the additional overhead of using synchronous mode will result in longer elapsed times with no additional benefits.

- As noted several times in this section, it will be important to monitor your resource utilizations on the source and target systems both prior to and after implementing remote journal. Key resources to consider include CPU utilization, line utilization, IOP utilizations, and DASD arm performance, as well as overall response times. These items can be monitored effectively using data collected from the performance monitor tool.

**Additional Sources of Information**

Customers interested in using the remote journal function should read the chapter on remote journal in the V4R2 version of the *OS/400 Backup and Recovery Guide*, SC41-5304. This chapter contains additional information on functional and performance aspects of remote journal.

## Parallel Index Build

Parallel index build, introduced in V4R1, has several important uses. This function can be used when the DB2 for AS/400 SMP feature is installed and made active. Following is a list of its uses.

- Speeding up the process of building a temporary index, when the query optimizer cannot locate an appropriate index and determines to build a temporary index to execute a particular query. This function is most beneficial in data warehousing environments housing large database files.

- Speeding up the process of building permanent indexes, particularly when built over large database files. Tests at the AS/400 Teraplex Center show that parallel index build, after the loading is complete,

is the fastest way to load data and build an index, and is preferred to building indexes while data is being loaded.

V4R3 note: In V4R3, the new encoded vector indexes, as well as the traditional binary radix tree indexes, are supported by the parallel index build function.

---

## 4.3 Version 3 DB2 for AS/400 Performance Information

Although much of the information in this section is still applicable to DB2 for AS/400, there may be portions of the articles that are no longer accurate due to changes made since the article was written. As of V4R3, the following general comments should be considered when using the information in this section.

- Prior to V4R1, *MAX4GB was the default value when creating any new index (as discussed below in "Enhanced Index Support for DB2/400" on page 43). Starting in V4R1, the default has been changed to *MAX1TB. In addition, the potential performance concerns with *MAX1TB indexes have been alleviated such that *MAX1TB is now the recommended size for all indexes (hence the change to the default value). The only concern with using *MAX1TB is that indexes created with this value can be up to 15% larger in size than if *MAX4GB is used, so it may be best to monitor available DASD space when converting or creating these types of indexes.

- In addition to the SQL improvements discussed in sections "DB2/400 SQL and Query Information Improvements" on page 44 and "DB2/400 SQL" on page 46, there have been additional changes made in subsequent releases that in many cases will allow well-tuned SQL applications on the AS/400 to be competitive with these same applications run on other hardware platforms. Although there are too many changes to list or discuss here, it is recommended that customers currently running their SQL applications on V4R1 or prior releases consider moving to V4R2 or a later release to take advantage of these performance improvements. Users should be able to take advantage of these improvements with no changes to or recompiles of existing SQL applications.

- In the "DB2/400 SQL" on page 46 section, several of the additional sources of information listed may no longer be available under the same document numbers or titles. The titles listed in this section were accurate as of V3R6 and V3R7, but may have since been moved or deleted. It may be best to contact your IBM representative if you need to know what current sources of information are available concerning SQL performance.

**DB2/400 Enhancements in V3R6**

**Enhanced Index Support for DB2/400**

The index support was enhanced in V3R6 to support larger indexes and to reduce index seize contention. Now rather than a size limitation of 4 Gigabytes (GB), each index can be as large as One Terabyte (TB). This limit is related to the size of the index and not the number of entries.

To take advantage of the increased index size capabilities, a new parameter, ACCESS PATH SIZE, has been added to the Create Physical File (CRTPF), Create Logical File (CRTLF), and Change Physical File (CHGPF) commands. If you want to allow growth beyond 4 GB on a keyed physical file that already exists, you can use the Change Physical File (CHGPF) command specifying "*MAX1TB" for the

ACCESS PATH SIZE parameter. If you want to change a logical file to the larger limit, you would delete the existing logical file and create a new logical file specifying "*MAX1TB" for the ACCESS PATH SIZE parameter. When creating new files, physical or logical, the default is *MAX4GB. You must specify *MAX1TB if it is required. It is recommended, though not required, that all access paths for a file be of the same type.

**Note**: The procedure described above will not work for files that have UNIQUE *YES specified. For these files you will have to create a second file with the ACCESS PATH SIZE specified at *MAX1TB and copy the first file to the second. The first file can then be deleted and the second file renamed.

In addition to the capability to create larger files, the algorithm used for seizing indexes has been enhanced. Indexes are now seized at the index page level. Therefore, for those workloads where there is a high-level of concurrency on a particular file or access path, the new algorithm will significantly reduce the contention resulting in significant performance improvements. This is particularly applicable to multi-processor systems and indexes with a high number of records being added. In order to take advantage of this change, the index must be created/changed as specified above.

Most customers will be best served by specifying the default of *MAX4GB for the ACCESS PATH SIZE parameter. This will in general provide better performance. Also if files will be moved to a prior release, the index may need to be rebuilt or the save of the file will not work if the index was created with *MAX1TB.

It is recommended that *MAX1TB be specified only where it is needed to allow for the larger file size or where there is high contention on access paths. If an index is changed to *MAX1TB and there is not high contention on the index, it may result in additional overhead. One measurement on a uni-processor system where this was the case resulted in a slight (less than 5%) slowdown. The amount of overhead would depend on, but not limited to, number of files, size of files, and access patterns.

**How to determine when to switch for performance**

To determine if the contention on your system is at a level that changing to a *MAX1TB access path will improve performance you will need to collect data using the Performance Monitor. This can be collected using the command STRPFRMON (Start Performance Monitor). When issuing this command, the interval value should be changed from 15 minutes to 5 minutes and the trace data collected should be changed from *NONE to *ALL. Data should be collected for at least 30 minutes and should be collected during "peak" activity. Please note that when the Performance Monitor is ended the trace data will automatically be dumped to DASD and performance could be degraded while this data is being transferred. This can be avoided by sepcifying Dump Trace *NO when starting the monitor and at a later time, when the system is not as busy, issuing the DMPTRCDTA command.

If the number of Seize Conflicts Per Second on the Component Report (created by issuing the PRTCPTRPT command or using the menu options from the PERFORM menu) is greater than 140 then you can in many cases benefit from changing to files with the ACCESS PATH SIZE parameter set to *MAX1TB. The method to determine which files should be changed is discussed in the next paragraph.

Once the trace data has been saved (either when the monitor ends if Dump Trace *YES was specified or after issuing the DMPTRCDTA command) a transaction report should be printed by either issuing the PRTTNSRPT command or using the menu options from the PERFORM menu. The SUMMARY OF SEIZE/LOCK CONFLICTS BY OBJECTS section of this report shows the number of seize conflicts and

the number of lock conflicts by object. This list will show conflicts on both data spaces (ie. file data) and data space indexes (ie. access paths). If the majority of seize conflicts are on one or two data space indexes, those are the candidates for the larger access path size parameters. The files most likely to fall in this range are those that are accessed by different users at the same time, some for inserting into the file, some for updating records in the file and some for reading the records in the file.

**Note**: For more detail on how to perform the functions above and what the reports contain as well as how to interpret the data, please consult the PERFORMANCE TOOLS/400 GUIDE publication, SC41-4340-00.

**DB2/400 SQL and Query Information Improvements**

In V3R6, there have been several enhancements made to improve performance for SQL queries. In addition, more information is now available to help users analyze performance for all queries.

1. In V3R6, there is now more information provided to query users to help them analyze and improve the performance of their queries:

   - Index advisor messages have now been added to the messages that are generated in the joblog when running a query in a job in DEBUG mode. These messages will indicate how an index could be constructed that would be optimal for the performance of that query. Note that the information provided is generally most useful for queries that involve a single file or for the primary file in a join query.

   - There is a new database performance monitor function available via the STRDBMON (Start Database Monitor) command. This monitor will provide detailed information on all DB2/400 queries, such as CPU, I/O, elapsed time, description of the query, etc. The information is placed in a database file where it can be readily queried. The data provided by this function can provide valuable information for performance analysis of any DB2/400 query. For more information on the database monitor, refer to the *DB2 for OS/400 Database Programming* guide.

2. Support has been added to allow more types of join operations such as outer joins and exception joins. Although users could previously construct queries involving UNIONs to do outer joins, this was often cumbersome to do. The new SQL join syntax now gives users the ability to easily code these types of queries, often with significantly improved performance versus the previous alternative methods. For more information on the new syntax, refer to the *DB2 for OS/400 SQL Programming* guide, (SC41-4611).

3. The new join syntax for SQL now also gives users the ability to specify the order in which they want files to be joined. This can help improve performance for join queries where the optimizer is not choosing the optimal order in which to do the join. More information on this is available in the above mentioned SQL Programming guide.

4. Prior to V3R6, all SQL cursors within a given program/module operated under the commitment control level specified for that program when it was compiled. Now, however, a new WITH clause has been added to the SQL SELECT statement to allow specific cursors to run under the desired level of commitment control. For example, if an SQL program is running with a commit level of *ALL but there are read-only cursors in the program that do not need any commitment control, you

can add the WITH NC clause to the SELECT statements to have these cursors run under a level of *NONE. Other levels that can be specified are UR (for *CHG), CS (for *CS) and RS (for *ALL). In many cases, a significant improvement in performance can be realized when this type of change is made to cursors that previously had been running under a more stringent level of commitment control than was necessary.

5. Support has been added to the ALTER TABLE SQL command that gives users the ability to easily add new fields and delete/change existing fields in any database file. Although users could do this in the past by deleting the old file and recreating it with a new format, this also meant that any views and indexes over the file had to be rebuilt as well, which could take a long time to complete. With the new support, the existing database file is copied to another file with the new format, and the indexes over the file are not rebuilt as long as no key fields are being altered. Although it still may take a while to do this copy, altering a database file's format with ALTER TABLE should be considerably faster than what the user had to do previously.

6. Prior to V3R6, a UNION ALL operation that did not specify an ORDER BY always generated a temporary file containing the results from each of the SELECT statements. This also meant that the ODP for this UNION was not reusable, which resulted in a full open and close each time the UNION was run. In V3R6, this type of operation now operates with 'live' data, i.e., the results from the second SELECT are not read until all the rows from the first SELECT have been read. This change will in most cases result in significantly improved response times for the first rows returned from the UNION since these rows can be returned immediately without having to wait for the entire temporary file to be filled with results from all the SELECTs. Also, SQL is now able to make the ODP for this operation reusable, which also improves response time significantly.

7. In previous releases, the ODP for a cursor that contained a LIKE clause with a host variable mask was not reusable, which meant a full open was required each time the query was run. In V3R6, the ODP will be reusable if the value in the host variable mask is of the form 'XXXX%' and the NUMBER of constant characters in the mask stays exactly the same between each run of the query. In the case shown here, the contents of the XXXX constant part may be changed, but the number of constant characters (4) must remain the same and there cannot be anything else in the mask. If these rules are adhered to, users can significantly improve the performance of this type of SQL query.

8. Queries such as UNIONs, subqueries and joins may in some cases specify the same view in each SELECT specified in the query. Prior to V3R6, running these types of queries resulted in the view being evaluated multiple times, once for each time it was specified. In V3R6, the view in cases like this is evaluated only once and the results are used by each SELECT in the query. This change can result in a noticeable improvement in performance for this type of query, particularly if each evaluation of the view is costly.

## DB2/400 SQL

DB2/400 Structured Query Language (SQL) support provides the user with an additional means of accessing data within an OS/400 relational database. This support provides several advantages in terms of flexibility, productivity and portability between various database platforms. However, prior to using SQL, users should also consider what level of performance they can expect when using this product. This section will provide general information on the performance of SQL to help users better determine what this level of performance will be.

This section is not intended to be a complete guide to SQL performance. For many users, other items of interest will include the SQL optimizer, performance tips and techniques, database design, etc. It is important that users take advantage of SQL performance tips and techniques as much as possible when writing an application using SQL. In particular, it is very important to properly construct and use indexes to provide the best overall performance for SQL. This, along with many other tips and techniques, can be found in the sources listed in “Additional Sources of Information” on page 48.

In V3R6, enhancements have been made to SQL that will help many users obtain an overall performance improvement. For a description of these enhancements, refer to “DB2/400 SQL and Query Information Improvements” on page 44.

**Performance of SQL versus Native DB**

When current users of AS/400 native language I/O (i.e., COBOL/400, RPG/400, etc.) are considering using SQL in their applications, one key item that needs to be considered is what level of difference in performance to expect when making this change.

Note that this section will not provide detailed information on the difference in performance between native and SQL for specific types of I/O operations. However, this type of information can be found in Chapter 5 of the document entitled *"SQL/400 - A Guide for Implementation"* (GG24-3321-01). Also, Chapter 3 of the second version of this document (GG24-3321-02) contains a section entitled "When to Use SQL" that provides additional information and guidelines on this subject.

It is difficult to predict how SQL will compare to native DB access for a given application. Generally, SQL will use anywhere from 10-30% more CPU than native, although this may vary considerably depending on the type of operations being done. For example, SQL shows considerably more overhead than native when operating on one record at a time, such as an OPEN-FETCH-UPDATE WHERE CURRENT OF-CLOSE sequence. However, for more complex operations or for operations involving a lower number of SQL statements, SQL will in many cases show relatively equal or better performance levels than native.

Note that the difference between SQL and native performance is usually in terms of CPU (the amount of I/O that occurs is generally about the same for native and SQL for similar functions). Note, however, that on systems where the CPU utilization has not reached the knee of the performance curve, a difference in CPU of 10-30% per transaction will not result in a large difference in response time. Beyond the knee, however, the response time difference may grow considerably.

**Other Performance Notes**

- Generally, the use of SQL will result in significantly increased memory requirements when compared to similar native DB operations. This is mainly due to the additional internal structures and program automatic storage required by SQL to maintain optimum performance levels, as well as the fact that SQL cannot share ODPs across or within applications as native DB can. However, it is important to remember that the extra memory required can vary widely from application to application and is mostly dependent on the complexity of the application. Simple applications involving only a small number of I/O operations may require little additional memory for SQL, but for complex applications involving many I/O operations the difference in memory requirements between native and SQL can be significant. When using SQL, users should monitor memory utilization using the AS/400 Performance Tools to better determine if additional memory will be required. In addition, support is available

through the Quicksizer tool on HONE to help users size their system for SQL.

- Some SQL users may notice an increase in the amount of auxiliary storage used once their applications begin running. The main reason for this is again largely due to the number and complexity of the ODPs and other internal structures that are maintained by SQL for each individual user for optimum performance. It is important to remember that this additional storage requirement is only temporary, i.e., when the user's job ends, the storage will return to normal levels. However, for systems where auxiliary storage usage is already high, some evaluation of the number of active SQL users and their storage requirements may be needed prior to any large scale implementation of a given SQL application.

- If memory and/or auxiliary storage is a concern, there are ways to reduce consumption of these resources for SQL applications. Following are some methods that can be used:

  - In a given SQL program, combine any duplicate or like SQL statements into one statement in an internal procedure, and then call that procedure as needed. This will help reduce the number of ODPs for that program.

  - If an internal procedure containing SQL statements is duplicated in several different SQL programs, the number of ODPs can be reduced by placing this internal procedure into a separate SQL program and then calling that program as needed. However, the user needs to be careful when doing this to ensure that the ODPs in this common program are reusable across different invocations of the program. Also, since external calls will cause some performance degradation, this also needs to be considered prior to implementing this type of change.

  - Some user applications "pre-open" all their SQL ODPs in order to avoid full opens when the SQL statements are issued. Although this will provide good performance in many cases, there may be ODPs that are pre-opened but rarely used. If this is true, the user may want to be more selective about which ODPs are pre-opened and which are left to be opened when the SQL statement is first issued.

**Additional Sources of Information**

There are several other sources of information available that the user should obtain to gain a better understanding of SQL and OS/400 database operations, as well as understanding how to properly code SQL functions in order to optimize performance. Following is a list of these sources.

- *SQL/400 - A Guide for Implementation*, GG24-3321

  There are three versions of this document currently available (01, 02 and 03). All versions contain key information and concepts users attempting to gain a better understanding of SQL operations.

  This information is also available on HONE. Please refer to HONE item number RTA000011842.

- *Quicksizer* support for SQL

  Available on HONE to help users size their system for using SQL.

- *System Selection Guide*

  Available on HONE under the title "System Selection Guide". Hardcopy editions of the U.S. version and the worldwide version are also available.

- *DB2 for OS/400 SQL Programming,* SC41-4611

- *DB2 for OS/400 Database Programming*

- *SQL/400 Programmer's Guide,* SC41-9609

- *SQL/400 Reference,* SC41-9608

- *AS/400 Database Guide,* SC41-9659

- *New Products Planning Information (NPPI),* GA41-0007

- FTN broadcasts entitled "SQL Performance Technical Update"

  There are several versions of this broadcast currently available on videotape. Each version covers SQL improvements made in each of the past several releases.

- Classes and workshops available to help users understand SQL programming and OS/400 database design and coding. One such recommended workshop is the "SQL Performance Workshop".

- SQL presentations from U.S. and European COMMON conferences.

**Query Management**

OS/400 Query Management (QM) is the AS/400 implementation of the SAA Common Programming Interface (CPI) Query. It provides a common method for accessing data and reporting the results from a relational database across the different platforms allowed by SAA. QM is also a very powerful and flexible reporting tool that provides users with the ability to design and format printed reports that result from the processing of a query. Queries can be included in programs written in RPG, COBOL, or C language and also can be run from within CL programs, giving programmers flexibility in how they set up the environment.

As a general rule, QM queries perform noticeably slower than similar functions that are done via AS/400 Query or embedded SQL because of the additional CPU used by QM (disk I/O characteristics are similar to those of AS/400 Query and SQL). For example, generating large reports via QM in most cases is significantly slower than AS/400 Query, and transaction processing with QM compares poorly with other alternatives such as static SQL. The exception to this rule is in processing summary-only functions such as AVG, COUNT, MIN, MAX and SUM. For this type of function, QM offers equal or better performance than AS/400 Query and performance levels similar to those when using SQL. Also, QM is optimized toward producing and displaying the first screen of output, so response times for this type of activity may be better than that for generating reports from QM.

QM should generally not be considered as an end-user tool. However, the SQL/400 Query Manager is available as an end-user type of interface for QM. Since it sits on top of QM and uses QM support, performance for this product will be similar to that of QM.

In general, customers who are considering the use of OS/400 Query Management need to weigh the functional advantages and flexibility (particularly in the area of portability across various SAA platforms) against the overall performance level of this product. Doing this will help decide if QM is a viable alternative to other AS/400 query products.

**Referential Integrity**

In a database user environment, there are frequent cases where the data in one file is dependent upon the data in another file. Without support from the database management system, each application program that updates, deletes or adds new records to the files must contain code that enforces the data dependency rules between the files. Referential Integrity (RI) is the mechanism supported by DB2/400 that offers its users the ability to enforce these rules without specifically coding them in their application(s). The data dependency rules are implemented as referential constraints via either CL commands or SQL statements that are available for adding, removing and changing these constraints.

For those customers that have implemented application checking to maintain integrity of data among files, there may be a noticeable performance gain when they change the application to use the referential integrity support. The amount of improvement depends on the extent of checking in the existing application. Also, the performance gain when using RI may be greater if the application currently uses SQL statements instead of HLL native database support to enforce data dependency rules.

When implementing RI constraints, customers need to consider which data dependencies are the most commonly enforced in their applications. The customer may then want to consider changing one or more of these dependencies to determine the level of performance improvement prior to a full scale implementation of all data dependencies via RI constraints.

**Triggers**

Trigger support for DB2/400 allows a user to define triggers (user written programs) to be called when records in a file are changed. Triggers can be used to enforce consistent implementation of business rules for database files without having to add the rule checking in all applications that are accessing the files. By doing this, when the business rules change, the user only has to change the trigger program.

There are three different types of events in the context of trigger programs: insert, update and delete. Separate triggers can be defined for each type of event. Triggers can also be defined to be called before or after the event occurs.

Generally, the impact to performance from applying triggers on the same system for files opened without commitment control is relatively low. However, when the file(s) are under commitment control, applying triggers can result in a significant impact to performance.

Triggers are particularly useful in a client server environment. By defining triggers on selected files on the server, the client application can cause synchronized, systematic update actions to related files on the server with a single request. Doing this can significantly reduce communications traffic and thus provide

noticeably better performance both in terms of response time and CPU. This is true whether or not the file is under commitment control.

The following are performance tips to consider when using triggers support:

- Triggers are activated by an external call. The user needs to weigh the benefit of the trigger against the cost of the external call.

- If a trigger is going to be used, leave as much validation to the trigger program as possible.

- Avoid opening files in a trigger program under commitment control if the trigger program does not cause changes to commitable resources.

- Since trigger programs are called repeatedly, minimize the cost of program initialization and unneeded repeated actions. For example, the trigger program should not have to open and close a file every time it is called. If possible, design the trigger program so that the files are opened during the first call and stay open throughout. To accomplish this, avoid SETON LR in RPG, STOP RUN in COBOL and exit() in C.

- If the trigger program opens a file multiple times (perhaps in a program which it calls), make use of shared opens whenever possible.

- If the trigger program is written for the Integrated Language Environment (ILE), make sure it uses the caller's activation group. Having to start a new activation group every time the time the trigger program is called is very costly.

- If the trigger program uses SQL statements, it should be optimized such that SQL makes use of reusable ODPs.

In conclusion, the use of triggers can help enforce business rules for user applications and can possibly help improve overall system performance, particularly in the case of applying changes to remote systems. However, some care needs to be used in designing triggers for good performance, particularly in the cases where commitment control is involved.

## System-Managed Access-Path Protection (SMAPP)

### Description

System-Managed Access-Path Protection (SMAPP) offers system monitoring of potential access path rebuild time and automatically starts and stops journaling of system selected access paths dynamically in order to meet a specified access path recovery time.

The default system wide access path recovery time for SMAPP is 150 minutes. This means that SMAPP protects the system so that there will be no more than 150 minutes of access path rebuild time during an IPL after an abnormal termination. Users can easily alter this value through the EDTRCYAP (Edit Recovery for Access Paths) or CHGRCYAP (Change Recovery for Access Paths) commands. SMAPP takes over the responsibility of providing the necessary amount of protection. No user intervention is required as SMAPP will manage the entire journal environment.

For systems with user auxiliary storage pools (ASPs), the recovery time can be specified for each ASP rather than one number for the entire system. This granularity allows the users to specify recovery time according to the criticality of the data on these ASPs. However, it is not recommended to specify target access path recovery times for both the entire system and individual ASPs.

For more information on SMAPP, see the *Backup and Recovery - Advanced Book* (SC41-3305).

**SMAPP Impacts on Overall System Performance**

The overhead of SMAPP varies from system to system and application to application due to the number of variables involved. For most customers, the default value of 150 minutes will minimize the performance impact while at the same time providing a reasonable and predictable recovery time and protection for key access paths. For many environments, even 60 minutes of IPL recovery time will have negligible overhead. Although SMAPP may start journaling access paths, the underlying SMAPP support is designed to be much cheaper in terms of performance than explicit journaling support.

Note that as the target access path recovery time is lowered, the performance impact from SMAPP will increase. You should balance your recovery time requirements against the system resources required by SMAPP.

Although the default level of SMAPP protection will be sufficient for most customers, some customers will need a different level of protection. The important variables are the number of key changes and the number of unprotected access paths. For those users who have experienced abnormal IPL access path recovery longer than 150 minutes it is advisable to experiment by varying the amount of protection. Too much protection causes undue CPU consumption whereas too little protection causes undesirable IPL delay. Customers may need to decide on an optimum SMAPP setting by understanding their system requirements and experimenting to find what value meets these requirements.

There is some help for those who want to experiment. The component report produced by the licensed program *Performance Tools/400* has a database journaling summary. It has information that can help explain the effects of various SMAPP settings. This information is also available to all customers without this licensed program except it takes a little work to query the information (see the chapter titled *Collecting Performance Data* in the *Work Management Guide*).

Users may also experience more DASD usage if they are explicitly journaling their physical files and SMAPP starts journaling for the access paths to the same user journal. However, this increase may be lessened by using the RCVSIZOPT(*RMVINTENT) option on the CRTJRN or CHGJRN command. This will cause the system to remove internal entries used only for IPL recovery when they are no longer needed.

There will be some customer environments (such as those having a tight batch window) where no additional performance overhead can be tolerated. For these environments, it recommended that the SMAPP setting be changed to a much higher number or *NONE prior to the batch window and then changed back to the default/chosen value during transaction-heavy hours.

If ANY overhead at all cannot be tolerated, SMAPP can be turned off completely (special value *OFF). In this mode, there is no performance overhead, but there is also no idea of how exposed the system is. Also, to turn SMAPP back on, the system must be in a restricted state. Therefore, it is not advisable to turn SMAPP *OFF. The differences between SMAPP *NONE and SMAPP *OFF are:

- SMAPP *NONE allows SMAPP to monitor the system exposure without journaling access paths.

- You do not have to be in a restricted state to change from SMAPP *NONE to any other setting.

**Miscellaneous Notes**

1. SMAPP has no performance impact when you run applications with no access paths or those that do not make any key changes.

2. If SMAPP has a noticeable impact to performance, it will generally be in terms of increased CPU utilization and/or increased asynchronous IO activity. In most cases, SMAPP will have little effect on the the amount of synchronous IO.

3. The system starts to journal ALL access paths when SMAPP is set at *MIN (minimum access rebuild time during IPL or maximum protection). In some environments, the overhead of *MIN can result in a significant impact to overall system performance. For this reason, *MIN is not a recommended setting. If you have several small access paths that have many key changes, you are better off paying the small price of rebuilding them in the IPL following an abnormal termination (which is not frequent) than paying the runtime overhead of maximum SMAPP protection.

4. SMAPP and explicit journaling (of physical files and/or access paths) can coexist and are compatible with each other.

5. If SMAPP decides to journal an access path for a physical file that is currently not being explicitly journaled, SMAPP must journal both the physical file and the access path. The impact from this change can be noticeable to an application's performance. However, if SMAPP also decides to journal more access paths for the physical file, the added cost of journaling each additional access path will be less than the impact from journaling the first access path.

## Journaling and Commitment Control

This section provides performance information and recommendations for DB2/400 journaling and commitment control.

### Journaling

The primary purpose of journal management is to provide a method to recover database files. Additional uses related to performance include the use of journaling to decrease the time required to back up database files and the use of access path journaling for a potentially large reduction in the length of abnormal IPLs. For more information on the uses and management of journals, refer to the *AS/400 Backup and Recovery Guide*.

- The addition of journaling to an application will impact performance in terms of both CPU and I/O as the application changes to the journaled file(s) are entered into the journal. Also, the job that is making the changes to the file must wait for the journal I/O to be written to disk, so response time will in many cases be affected as well.

  Journaling impacts the performance of each job differently, depending largely on the amount of database writes being done. Applications doing a large number of writes to a journaled file will most

likely show a significant degradation both in CPU and response time while an application doing only a limited number of writes to the file may show only a small impact.

- The impact to performance from adding journaling can be reduced by locating the journal receiver on a separate user ASP. Doing this will generally reduce the seek time required to access the disk arms for journal I/O which will in turn help reduce the impact to end user response time. It will also lessen the impact to the disk arms located on the system ASP.

  When using a separate user ASP for journal receivers, it is important to consider the number of disk actuators available in the ASP. Customer environments with heavily used journal receivers located in a user ASP that consists of a single disk actuator may actually reach a limit to performance because of the high usage of this single actuator. In this case, it would be better to have multiple disk actuators available in the user ASP so that DB2/400 journaling support can interleave journal entries over the multiple actuators, thus reducing contention for any one single disk arm. Doing this may result in an improvement in response time and in overall system throughput. However, it is important to note that although adding an actuator may provide a significant improvement in performance, each additional actuator added beyond this will improve performance to a lesser degree. Once the utilization of the actuators is low, adding more actuators will not improve performance.

  Having two or more journal receivers located on the same user ASP and having them in use at the same time may not take full advantage of the performance gains seen by isolating a single journal receiver on the User ASP since the seek distance on the actuator increases as the journal entries are written to the two receivers.

- Tracked asynchronous I/O is used to write the journal entries to disk. The use of this type of I/O allows the journal support to determine on a process by process basis, which processes need to wait for the I/O to complete and which are allowed to continue. However, by using tracked asynchronous I/O, all I/O operations to a journal receiver now appear in performance reports as asynchronous even though the process may actually be waiting for the I/O operation to complete. This could cause the Capacity Planning tools to recommend a smaller configuration than is necessary. This should beconsidered if a measured profile is created for purposes of future system capacity planning.

**Commitment Control**

Commitment control is an extension to the journal function that allows users to ensure that all changes to a transaction are either all complete or, if not complete, can be easily backed out. The use of commitment control adds two more journal entries, one at the beginning of the committed transaction and one at the end, resulting in additional CPU and I/O overhead. In addition, the time that record level locks are held increases with the use of commitment control. Because of this additional overhead and possible additional record lock contention, adding commitment control will in many cases result in a noticeable degradation in performance for an application that is currently doing journaling.

- There are instances where adding commitment control can result in improved response times for an application doing journaling. As stated before, journaling alone means that the journal entries for changes to the file are written synchronously to disk. However, under commitment control, most journal entries are written to disk asynchronously. Only the final journal entry of the commit cycle (along with any entries of the cycle that have not yet been written to disk) are written synchronously. Because of this, applications may no longer have to wait for each journaled change to be written, which can result in reduced response times. The amount of improvement will depend mainly on the number of

journal entries within the commit cycles - the more entries per cycle, the greater the potential for improving response time over journaling alone. For example, adding commitment control to a dedicated batch job that is currently doing journaling could potentially improve the job run time if there are a large number of changes to the physical files being journaled.

- It is important to remember that the potential for improving response time by adding commitment control is also largely affected by overall system resource utilization. Environments that are showing high CPU or disk utilization or have constrained memory will in most cases show a degradation in performance from adding commitment control because of the additional CPU and I/O required. Also, adding commitment control can result in record level lock contention between jobs, which can also affect response time. Given the number of variables involved, a test run is highly recommended prior to adding commitment control for the purpose of improving performance in a production environment.

**Date/Time Fields**

The support for date and time fields in DB2/400 provides a number of advantages for the end user:

- Programmer productivity may be improved when an application requires calculations on date or time fields. New functions can be added more easily.

- Since the date and time data is stored in an internal format and converted on retrieval, the same underlying data can be viewed in different formats based on the needs of the application.

- Because the internal format reflects the sequential nature of time, it can be easily used to sort data in terms of sequence. For example, if a file currently contains a date in MMDDYY format, special application processing is required to sort it in YYMMDD sequence. This application processing is not needed when the date is stored in internal format.

- Some applications may achieve small savings in file size and DASD requirements since the internal formats are generally smaller than external formats.

The use of DB2/400 date/time support in many cases will result in additional CPU resource being used. Generally, the increase will be less than 10% but is dependent upon the number of calculations and the number of date and time fields being accessed. Time fields will usually show minimal impact while date and timestamp data types may show more of an effect on performance.

Note that in terms of performance, DB2/400 date/time support is generally better than or equal to other generalized routines that support many different date/time formats. However, when compared to date/time routines that handle only very specific date/time formats, DB2/400 date/time support may have higher CPU requirements.

When using date/time support in products such as AS/400 Query and Query Management, the amount of additional CPU required will vary. In many instances, the impact will be minimal and may even show a small reduction in CPU versus previous methods of providing this type of support. For example, report breaks on date fields under AS/400 Query will in many cases provide comparable performance to using packed data for dates. However, there are certain cases where the use of date/time support can result in significant performance overhead:

- When replacing the use of zoned decimal data for dates
- When adding a result field calculations to a query (such as adding 90 days to a date)
- Report breaks on date fields under Query Management (compared to the use of packed data for dates)

Overall, DB2/400 date/time support can provide many functional advantages to user applications without a significant impact to performance. However, the user should exercise some caution when implementing this support in order to minimize this impact.

**Null Values**

DB2/400 provides support for the use of null values in any field in any file. For a more detailed description of null value support, refer to the SQL/400 Reference or the SQL/400 Programmers Guide.

The performance impact from using null value support will vary depending on the number of fields declared as null capable and on the number of records being accessed. For example, when a user even changes only one field in a file to be null capable, there will be a slight increase in the CPU resource required to either insert records into or read records from this file. The amount of the increase should be about the same whether or not the null capable field actually contains null values. Also, as the number of null capable fields in a given file record format increases, the CPU required to process each record will also increase. For operations such as AS/400 Query, Query Management and SQL/400 queries that select all the fields from a large number of records, the impact of adding null capable fields to the file can be significant in terms of increased CPU.

Because of the potential impact, users need to be somewhat careful in what files null capable fields will be used and in deciding how many fields will be null capable. Although null capable fields do provide good functional advantages, performance also needs to be considered prior to using this support.

**CCSID Support**

CCSID (Coded Character Set Identification) enhancements support the dynamic conversion of data from one language to another. The support allows jobs, files, and fields within files to be tagged with an identification of the code page currently being used. For a more detailed description of this support refer to the AS/400 National Language Support Planning Guide.

The main effect to performance from CCSID support is from the character data conversion required when either the CCSID of the job and the file/field do not match or when either of these CCSID values is not set to 65535. The amount of additional CPU required for this conversion will vary somewhat depending on the amount of character data that needs to be converted. Since the impact of this conversion can be significant to normal database operations, users should exercise some caution when implementing this function. For example, it may be best to consider doing CCSID conversion only on fields that need the conversion done instead of all character data in the given database file.

**Sort Sequence**

DB2/400 sort sequence support provides application developers and end users with an easy method of producing sorted data for a particular language or culture. A set of unique and shared sort sequence tables

are included on the AS/400. Developers can refer to sort sequences when creating applications using database, Query/400, RPG, COBOL, C, and ILE/C compilers, as well as SQL precompilers.

The performance of sort sequence support should be compared to the alternative methods that users have available on the AS/400. For example, users who desire to use a different sorting sequence in QUERY/400 queries can create a translation table and then specify this translation table in the "select alternate collating sequence" option in QUERY/400. However, comparisons of these two methods show that sort sequence support will provide a noticeable improvement in performance (ranging from 5-40%) versus using the translation table method.

Users who would like to learn more about sort sequences should refer to the *National Language Support Planning Guide*.

## Variable Length Fields

Variable length field support allows a user to define any number of fields in a file as variable length, thus potentially reducing the number of bytes that need to be stored for a particular field.

### Description

Variable length field support on the AS/400 has been implemented with a spill area, thus creating two possible situations: the non-spill case and the spill case. With this implementation, when the data overflows, all of the data is stored in the spill portion. An example would be a variable length field that is defined as having a maximum length of 50 bytes and an allocated length of 20 bytes. In other words, it is expected that the majority of entries in this field will be 20 bytes or less and occasionally there will be a longer entry up to 50 bytes in length. When inserting an entry that has a length of 20 bytes or less that entry will be inserted into the allocated part of the field. This is an example of a non-spill case. However, if an entry is inserted that is, for example, 35 bytes long, all 35 bytes will go into the spill area.

To create the variable length field just described, use the following SQL/400 statement:

```
CREATE TABLE library/table-name
       (field VARCHAR(50) ALLOCATE(20) NOT NULL)
```

In this particular example the field was created with the NOT NULL option. The other two options are NULL and NOT NULL WITH DEFAULT. Refer to the NULLS section in the SQL/400 Reference Guide to determine which NULLS option would be best for your use. Also, for additional information on variable length field support, refer to either the SQL/400 Reference Guide or the SQL/400 Programmer's Guide.

### Performance Expectations

- Variable length field support, when used correctly, can provide performance improvements in many environments. The savings in I/O when processing a variable length field can be significant. The biggest performance gains that will be obtained from using variable length fields are for description or comment types of fields that are converted to variable length. However, because there is additional overhead associated with accessing the spill area, it is generally not a good idea to convert a field to variable length if the majority (70-100%) of the records would have data in this area. To avoid this problem, design the variable length field(s) with the proper allocation length so that the amount of data in the spill area stays below the 60% range. This will also prevent a potential waste of space with the

variable length implementation.

- Another potential savings from the use of variable length fields is in DASD space. This is particularly true in implementations where there is a large difference between the ALLOCATE and the VARCHAR attributes AND the amount of spill data is below 60%. Also, by minimizing the size of the file, the performance of operations such as CPYF (Copy File) will also be improved.

- When using a variable length field as a join field, the impact to performance for the join will depend on the number of records returned and the amount of data that spills. For a join field that contains a low percentage of spill data and which already has an index built over it that can be used in the join, a user would most likely find the performance acceptable. However, if an index must be built and/or the field contains a large amount of overflow, a performance problem will likely occur when the join is processed.

- Because of the extra processing that is required for variable length fields, it is not a good idea to convert every field in a file to variable length. This is particularly true for fields that are part of an index key. Accessing records via a variable length key field is noticeably slower than via a fixed length key field. Also, index builds over variable length fields will be noticeably slower than over fixed length fields.

- When accessing a file that contains variable length fields through a high-level language such as COBOL, the variable that the field is read into must be defined as variable or of a varying length. If this is not done, the data that is read in to the fixed length variable will be treated as fixed length. If the variable is defined as PIC X(40) and only 25 bytes of data is read in, the remaining 15 bytes will be space filled. The value in that variable will now contain 40 bytes. The following COBOL example shows how to declare the receiving variable as a variable length variable:

```
01 DESCR.
       49 DESCR-LEN      PIC S9(4) COMP-4.
       49 DESCRIPTION    PIC X(40).

   EXEC SQL
       FETCH C1 INTO DESCR
   END-EXEC.
```

  For more detail about the vary-length character string, refer to the SQL/400 Programmer's Guide.

  The above point is also true when using a high-level language to insert values into a variable length field. The variable that contains the value to be inserted must be declared as variable or varying. A PL/I example follows:

```
DCL FLD1 CHAR(40) VARYING;
FLD1 = XYZ Company;

EXEC SQL
    INSERT INTO library/file VALUES
        ("001453", FLD1, ...);
```

  Having defined FLD1 as VARYING will, for this example, insert a data string of 11 bytes into the field corresponding with FLD1 in this file. If variable FLD1 had not been defined as VARYING, a

data string of 40 bytes would be inserted into the corresponding field. For additional information on the VARYING attribute, refer to the PL/I User's Guide and Reference.

- In summary, the proper implementation and use of DB2/400 variable length field support can help provide overall improvements in both function and performance for certain types of database files. However, the amount of improvement can be greatly impacted if the new support is not used correctly, so users need to take care when implementing this function.

## Reuse Deleted Record Space

### Description of Function

This section discusses the support for reuse of deleted record space. This database support provides the customer a way of placing newly-added records into previously deleted record spaces in physical files. This function should reduce the requirement for periodic physical file reorganizations to reclaim deleted record space. File reorganization can be a very time consuming process depending on the size of the file and the number of indexes over it. To activate the reuse function, set the Reuse deleted records (REUSEDLT) parameter to *YES on the CRTPF (Create Physical File) or CHGPF (Change Physical File) commands. The default value when creating a file is *NO (do not re-use).

### Comparison to Normal Inserts

Inserts into deleted record spaces are handled differently than normal inserts and have different performance characteristics. For normal inserts into a physical file, the database support will find the end of the file and seize it once for exclusive use for the subsequent adds. Added records will be written in blocks at the end of the file. The size of the blocks written will be determined by the default block size or by the size specified using an Over-ride Database File (OVRDBF) command. The SEQ(*YES number of records) parameter can be used to set the block size.

In contrast, when re-use is active, the database support will process the added record more like an update operation than an add operation. The database support will maintain a bit map to keep track of deleted records and to provide fast access to them. Before a record can be added, the database support must use the bit-map to find the next available deleted record space, read the page containing the deleted record entry into storage, and seize the deleted record to allow replacement with the added record. Lastly, the added records are blocked as much as permissible and then written to the file.

To summarize, additional CPU processing will be required when re-use is active to find the deleted records, perform record level seizes and maintain the bit-map of deleted records. Also, there may be some additional disk IO required to read in the deleted records prior to updating them. However, this extra overhead is generally less than the overhead associated with a sequential update operation.

### Performance Expectations

The impact to performance from implementing the reuse deleted records function will vary depending on the type of operation being done. Following is a summary of how this function will affect performance for various scenarios:

- When blocking was not specified, re-use was slightly faster or equivalent to the normal insert application. This is due to the fact that reuse by default blocks up records for disk IOs as much as

possible.

- Increasing the number of indexes over a file will cause degradation for all insert operations, regardless of whether reuse is used or not. However, with reuse activated, the degradation to insert operations from each additional index is generally higher than for normal inserts.

- The RGZPFM (Reorganize Physical File Member) command can run for a long period of time, depending on the number of records in the file and the number of indexes over the file. Even though activating the reuse function may cause some performance degradation, it may be justified when considering reorganization costs to reclaim deleted record space.

- The reuse function can always be de-activated if the customer encounters a critical time window where no degradation is permissible. The cost of activating/de-activating reuse is relatively low in most cases.

- Because the reuse function can lead to smaller sized files, the performance of some applications may actually improve, especially in cases where sequential non-keyed processing of a large portion of the file(s) is taking place.

## DB2/SMP Feature

### Introduction

The symmetrical multiprocessing (SMP) feature provides additional query optimization algorithms for retrieving data. In addition, the DB2/SMP feature provides application transparent support for parallel query operations on a single tightly-coupled multi-processor AS/400 system (shared memory and disk). The database manager can automatically activate parallel query processing in order to engage one or more system processors to work simultaneously on a single query. The response time can be dramatically improved when a processor bound query is executed in parallel on multiple processors. The purpose of this section is to:

- Introduce new query optimization algorithms available with the DB2/SMP feature.
- Briefly discuss decision support (DSS) queries which will realize the most benefit with the SMP feature.
- Provide guidance to help estimate DSS query capacity on various AS/400 systems.

### New Query Optimization Algorithms

The DB2/SMP feature provides the following new query optimization algorithms:

- Parallel table scan

  Provides parallel operations for queries requiring a sequential scan of the entire table. Multiple tasks are used to scan the same table concurrently. Each task will perform selection and column processing on a table partition and return selected records to the requester. The response time improvement for a parallel table scan scales closely to the number of processors participating. For example, the response time for a table scan can be up to 4 times faster when run in parallel on a 4-way processor.

- Index only access (parallel and non-parallel)

Provides performance improvement by extracting a query answer from an index rather than performing random I/Os against a physical table. For this to happen, all columns that are referenced in a query must exist within an index. Response time improvements can be up to 5 times faster for some queries.

- Parallel key selection

  Provides parallel index operations for key selection. Multiple tasks are used to scan the same index concurrently. Each task will search a different key range and selected records are returned to the requester.

- Hashing algorithms

  Provides an optimization alternative for group by and some join queries. This method avoids having to utilize an index and therefore avoids having to perform random I/Os to retrieve the results. Instead, a temporary partitioned hash table can be used. This table can be processed by large and efficient sequential I/Os and often utilizing parallel table scan to provide the results. Response time improvements for group by queries can be up 6 times better and some joins can be up to 25 times improved (4 to 10 times is more typical).

The SMP feature was available V3R1 on AS/400 IMPI models and became available V3R7 for AS/400 RISC models. For more information on the SMP feature and the new algorithms, see *TNL SN41-3680 to SC41-3611-00*.

**Decision Support Queries**

The SMP feature is most useful when running decision support (DSS) queries. DSS queries generally give answers to critical business questions tend to have the following characteristics:

- examine large volumes of data
- are far more complex than most OLTP transactions
- are highly CPU intensive
- includes multiple order joins, summarizations and groupings

DSS queries tend to be long running and can utilize much of the system resources such as processor capacity (CPU) and disk. For example, it is not unusual for DSS queries to have a response time longer than 20 seconds. In fact, complex DSS queries may run an hour or longer. The CPU required to run a DSS query can easily be 100 times greater than the CPU required for a typical OLTP transaction. Thus, it is very important to choose the right AS/400 system for your DSS query and data warehousing needs.

**SMP Performance Summary**

The SMP feature provides performance improvement for query response times. The overall response time for a set of DSS queries run serially at a single work station may improve 25 to 58 percent when SMP support is enabled. The amount of improvement will depend in part on the number of processors participating in each query execution and the optimization algorithms used to implement the query. Some individual queries can see significantly larger gains. Queries that are able to utilize the new hash join algorithm may see up to a 25 times improvement in query response time. In addition, query throughput may improve 18 to 25 percent because the new optimization algorithms require less CPU resource. The new hashing algorithms also dramatically reduce the number of disk IOs.

## Capacity Planning

The Capacity Planning sections contain the following information:

- Initial system sizing recommendations for data warehouses
- Detailed capacity planning information for various AS/400 models. This information will be useful when you are able to determine a customer's average DSS query response time and want to compare running a query workload on other AS/400 models or with the SMP feature enabled.
- Capacity planning tips

### System Sizing Recommendations for Data Warehouses

The following table gives some high-level guidance for choosing the AS/400 system for Data Warehouses based on the size of the database and/or the maximum number of concurrent users.

*Table 16. System Sizing Recommendations for AS/400 Data Warehouses*

| System | Maximum Data in Gigabytes | Maximum Number of Concurrent Users |
|---|---|---|
| 40S | 15 | 15 |
| 50S 2120 | 220 | 20 |
| 50S 2121 | 220 | 30 |
| 53S 2154 | 350 | 40 |
| 53S 2155 | 350 | 65 |
| 53S 2156 | 350 | 125 |

Note:
1. The maximum amount of main storage exists on each system.
2. Query workloads are assumed to be comprised of the following query mixture:
   - Simple queries (no joins or group by aggregation) 80%
   - Medium queries (2-way joins, group by aggregation) 15%
   - Complex queries (union, subselects) 5%
3. Simple query workloads may also include the use of any Multi-Dimensional Database product.
4. If your database size is greater than 350 Gigabytes or the number of concurrent users is greater than 125, you will require a multisystem implementation which may include the DB2 Multisystem feature. DB2 Multisystem provides the capability of horizontally partitioning a table across multiple systems and running a single query in parallel. Performance is improved due to parallel operations and because of the table partitioning. Each system needs only scan a fraction of the entire database when a query is run.

### Capacity Planning based upon Average Query Response Time

This section provides some guidance to help you estimate DSS query capacity on various AS/400 systems, with and without the SMP feature enabled. The chart was developed based upon studying the results of various customer and synthetic DSS workloads. The workloads contained various sized files ranging from 25 records up to 100 million records. A broad range of DSS support queries, from simple to complex were measured. Queries that utilized joins, group by, and summarizations were commonplace. The database structure, the index structure, and the query syntax are all assumed to be optimal. The SMP numbers in the the chart show a range of performance based upon an estimate of:

- the percentage of the DSS queries that might be helped by SMP
- the query benefit provided by parallelism

### How to use the chart

Calculating the capacity for DSS query workloads can be difficult due to vast variability of the queries. The capacity chart uses an average query response time that might be observed in a customer environment over a long period of time such as a day. Obviously, during this time period there will be great variance in terms of complexity of the queries, the size of the tables queried, the response times of the individual queries, and the load put on the AS/400 system.

For the capacity chart, we have used 180 seconds to represent a customer's average query time. An average of 180 seconds would indicate a majority of simple DSS queries being executed during the time window. If this does not accurately reflect your customer's environment, you can estimate new average response times and system capacities by performing the calculations that follow the capacity chart:

*Table 17. System Capacity Planning - DSS Queries*

| Model | CPUs | Avg Response Time in Seconds | | Capacity in Queries/Hour | |
|---|---|---|---|---|---|
| | | No SMP | SMP Range | No SMP | SMP Range |
| 30S 2411 | 1 | 180 | -- | 40 | -- |
| 30S 2412 | 2 | 173 | 93-130 | 86 | 105-114 |
| E95 | 4 | 188 | 78-117 | 148 | 179-195 |
| F97 & 320 | 4 | 154 | 64-96 | 225 | 272-297 |
| 50S 2121 | 1 | 100 | -- | 98 | -- |
| 53S 2154 | 1 | 86 | -- | 159 | -- |
| 53S 2155 | 2 | 86 | 46-64 | 244 | 296-324 |
| 53S 2156 | 4 | 86 | 36-54 | 372 | 451-493 |

**Note**:
1. The average CPU reduction when SMP is enabled 18-25%.
2. Capacity numbers are based on 100% CPU utilization and assume that the system is dedicated to query processing.
3. Information in chart based on assumptions listed in the next section.

You can estimate new average response times and system capacities by performing the following calculations against the values in the capacity chart:

- Determine the customers average response time
- Compute the following query response time ratio:
  ratio = customer's average response time/average response time from table
- Multiply all response times by the ratio to get new response times
- Divide all capacities by the ratio to get new capacities
- For example, if the customer's average response time is 38.5 seconds on an F97 processor, to calculate the new F97 values, perform the following calculations:

```
Current F97 row:  F97 & 320  4   154   64-96     225   272-297

                  Ratio = 38.5/154 = .25

                  Response times      Capacities
                  ------------------------------------
                  154 (*.25) = 38.5     225/.25 =  900   No SMP
                   64 (*.25) = 16.0     272/.25 = 1088   SMP
                   96 (*.25) = 24.0     297/.25 = 1188   SMP
New F97 row:      F97 & 320  4   38.5  16-24     900   1088-1188
New 53S 2156 row: 53S 2156   4   21.5   9-13.5  1488   1804-1972
```

### Capacity Planning Assumptions

The following assumptions were used to help generate the capacity chart:

1. DSS query workloads can be characterized by an average response time. The average response time will increase as the size of the customer's database size increases.
2. Given all of a customer's DSS queries, typically 50%-70% of the queries will utilize the SMP support.
3. For queries that utilize SMP, the response time will scale relative to the number of CPUs. The scaling range equals 1 to 1.5 times the number of CPUs involved in the query execution. For example, on a 4-way CPU system, response time will be 1/4 to 1/6 the time compared to executing the query on just one of the processors.
4. Hash group by/join algorithms will be utilized in about 70% of all queries that can utilize the SMP support. About 50-75% of the IOs will be eliminated when the new hashing algorithms are used.

5. Table scans will be utilized in about 10% of all the DSS queries.
6. For SMP queries, CPU consumption will decrease up to 35% due to the new optimization algorithms and because of the reduction in disk Ios.
7. A DSS query workload will utilize at least 50% of the system processor capacity when run on an AS/400 30S 2411.

**Capacity Planning Tips**

Here are some suggestions that may improve your DSS query performance when utilizing the SMP support:

- Add additional memory. 20%-25% of the active database should reside in main memory.

- Add additional disks and limit the the number of disks per controller to 8 if possible. This is especially true if you are using 9337 DASD and the 6501 DASD IOP, as this will cause more efficient use of active memory.

- Utilize fast DASD (6503,6506,and 6507) and DASD IOPs (6502 and 6512). Spread the IOPs evenly among the system busses.

- Ensure that the database is spread evenly over multiple DASD arms. Installing DASD that is all the same size helps ensure even spreading.

- For smaller database sizes (< 30GB), you should have 2-3 DASD arms per CPU to get good performance.

- Utilize RISC hardware. RISC systems have faster system busses with larger bandwidths than those found on IMPI systems. In addition, disk IO sizes are larger which will results in fewer disk Ios.

- Be sure that there is enough space on the the system auxiliary storage pool (ASP) to allow the database manager to create temporary files for query execution. Do not exceed 70% capacity on the system ASP.

- Under a heavy system load, limit the amount of query parallelism. The degree of parallel activity can be controlled by the user via the CHGSYSVAL (parm QQRYDEGREE) and CHGQRYA (parm Degree) CL commands.

## DB2 Multisystem for OS/400

DB2 Multisystem for OS/400 offers customers the ability to distribute large databases across multiple AS/400s in order to gain nearly unlimited scalability and improved performance for many large query operations. The multiple AS/400s are coupled together in a shared-nothing cluster where each system uses its own main memory and disk storage. Once a database is properly partitioned among the multiple nodes in the cluster, access to the database files is seamless and transparent to the applications and users that reference the database. To the users, the partitioned files still behave as though they were local to their system.

This section will provide information on what level of performance improvements to expect from DB2 Multisystem as well as tips and techniques on how to install and use this product for optimal performance.

However, this section should not be viewed as a complete guide to performance for DB2 Multisystem. It is recommended that in addition to the information provided here, you should obtain the following documents to help understand more about both the key performance and functional aspects of this product.

- *DB2 Multisystem for OS/400,* SC41-3705-00

  This document is an excellent overall reference for this product and contains several aspects of performance that will not be covered in this document, in particular some items on distributed query optimization and processing.

- *Slash DB2/400 Query Time with Parallel Processing*

  This article (found in the April 1996 edition of the NEWS/400 magazine) helps explain key performance and functional concepts of DB2 Multisystem.

These documents and the information in this section assumes that you are familiar with nondistributed query performance on the AS/400 and that you have a good overall background in database concepts. Other documents that can help you with this information include:

- *DB2 for OS/400 SQL Reference*

- *DB2 for OS/400 SQL Programming*

- *DB2 for OS/400 Database Programming*

- *CL Reference Guide*

**Planning for DB2 Multisystem**

The most important aspect of obtaining optimal performance with DB2 Multisystem is to plan ahead for what data should be partitioned and how it should be partitioned. The main idea behind this planning is to ensure that the systems in the cluster run in parallel with each other as much as possible when processing distributed queries while keeping the amount of communications data traffic to a minimum. Following is a list of items to consider when planning for the use of distributed data via DB2 Multisystem.

- Avoid large amounts of data movement between systems. A distributed query often achieves optimal performance when it is able to divide the query among several nodes, with each node running its portion of the query on data that is local to that system and with a minimum number of accesses to remote data on other systems. Also, if a file that is heavily used for transaction processing is to be distributed, it should be done such that most of the database accesses are local since remote accesses may add significantly to response times.

- Choosing which files to partition is important. The largest improvements will be for queries on large files. Files that are primarily used for transaction processing and not much query processing are generally not good candidates for partitioning. Also, partitioning files with only a small number of records will generally not result in much improvement and may actually degrade performance due to the added communications overhead.

- Choose a partitioning key that has many different values. This will help ensure a more even distribution of the data across the multiple nodes. In addition, performance will be best if the partitioning key is a single field that is a simple data type.

- It is best to choose a partition key that consists of a field or fields whose values are not updated. Updates on partition keys are only allowed if the change to the field(s) in the key will not cause that record to be partitioned to a different node.

- If joins are often performed on multiple files using a single field, use that field as the partitioning key for those files. Also, the fields used for join processing should be of the same data type.

- It will be helpful to partition the database files based on how quickly each node can process its portion of the data when running distributed queries. For example, it may be better to place a larger amount of data on a large multiprocessor system than on a smaller single processor system. In addition, current normal utilization levels of other resources such as main memory, DASD and IOPs should be considered on each system in order to ensure that no one individual system becomes a bottleneck for distributed query performance. For information on how to customize your database partitioning, refer to the "DB2 Multisystem for OS/400" document mentioned above.

- For the best query performance involving distributed files, avoid the use of commitment control when possible. DB2 Multisystem uses two-phase commit, which can add a significant amount of overhead when running distributed queries.

In addition to these items, the document and article referenced above contain other key concepts that should be considered while planning your data distribution via DB2 Multisystem.

**Performance During Data Distribution**

Generally, partitioning large database files across multiple systems can be a long process during which the data in the files is unavailable. Following is a list of items that should be considered prior to actually partitioning the files to help reduce the time this process may take.

- The use of Opticonnect will result in significantly better distribution times than using other alternatives such as a 16Mbps Token Ring LAN. Opticonnect will also help improve performance for distributed queries that result in large amounts of data being moved from node to node to complete the query.

- There are basically two recommended methods of distributing data from a local system to a set of systems linked together with DB2 Multisystem. One method is to use the Change Physical File (CHGPF) command with the NODGRP and PTNKEY parameters. This command will need to be issued against each database file to be distributed. Any existing logical files for this file will also be rebuilt on a per node basis. The second method is to create a new physical file with the same data format as the original and with the node group and partition key specified (this can be done either via the Create Physical File (CRTPF) command or the SQL CREATE TABLE command), and then issue a Copy File (CPYF) command to copy the data from the original file to the new distributed file. Measurement results show that the performance of these methods is about equal.

  Note that there is a faster and slower version of both the CHGPF and CPYF operations for distributing files. The faster version sends large buffers of records at a time while the slower version sends one record at a time. To see if the fast version is being done, look for occurrences of the CPC9203 message

in the joblog of the job doing the distribution, stating how many records were copied to each node. If these messages do not appear, the slower version is being used. The factors that influence which version is used are listed in more detail in the DB2 Multisystem for OS/400 document mentioned above.

- To help the distribution process, it may be best to keep the number of logical files to a minimum for the physical files that are being distributed. These logical files can then be built via the Create Logical File (CRTLF) or the SQL CREATE INDEX command at a later time, possibly in background batch jobs. This approach is generally faster than having the system maintain or build the indexes on each node as the physical data is distributed. However, you will have to issue the index builds separately and they will tend to cause high CPU utilization while they are occurring, so this must be considered as well. If you need certain key indexes to exist as soon as the data distribution is done, you should let the CPYF or CHGPF operations handle these for you.

- It will be to your benefit to avoid the use of commitment control or journaling while distributing database files. The use of these options will add significantly to the overall distribution time.

- The time for data distribution may also be helped by having several jobs running at the same time, each distributing a different file. Although this is best accomplished where the system doing the distribution is a multiprocessor system, this can also apply to single processor systems. The key to making this work is to avoid a bottleneck on a resource such as main memory, DASD, CPU or the communications lines or IOPs. It may be best to try this by adding one job at a time and monitoring system performance to see if any resources are becoming overutilized.

**Distributed Query Performance**

The performance of queries run over distributed data will in many cases improve significantly compared to the performance that had been achieved running these same queries on a single system. However, there also may be queries that show little or no performance gain, with some possibly showing degradation in performance. The following information should help determine what level of performance to expect when running queries over distributed data. Again, it is important to reference the above mentioned documents in conjunction with the information provided here in order to gain a more complete picture of distributed query performance.

- Use of the new ASYNCJ parameter on the Change Query Attributes (CHGQRYA) command is very important to achieving the best performance levels for distributed queries. The value specified for this parameter will greatly affect the response time for distributed queries by altering the degree of parallelism allowed as well as the amount of work done by the temporary result writer jobs. Note that this command needs to be issued on a per job basis as there is no global system level value that can be changed. For more information on the use of this command for distributed queries, refer to the *DB2 Multisystem for OS/400* document.

- There is now a distributed query optimizer that operates only on distributed queries. This optimizer determines what steps are necessary to efficiently run the distributed query and what nodes will process these individual steps. Local level optimization on each node is still handled by the previously existing query optimizer.

- The use of Opticonnect is recommended for the best overall performance of distributed queries. Although good planning will minimize the amount of communications overhead needed for many

distributed queries, there still will be a fair amount of cross-system data traffic in many DB2 Multisystem environments. Using Opticonnect will result in noticeably better response times for queries with a significant amount of cross-system data movement and will in general help reduce the communications overhead for users of DB2 Multisystem.

- Generally, the best performance gains from DB2 Multisystem will be for queries that exhibit the following characteristics:

  - ❖ The query processes a large number of records
  - ❖ The query can be divided such that subsets of the records it processes can be queried on multiple nodes in parallel
  - ❖ Each part of the divided query returns a small number of records to the coordinating system where the query originated

  For queries that meet these criteria, performance can be expected to improve in nearly a linear progression with the number of systems involved in running the query. In addition, if any of the systems used are multiprocessor systems, the improvement on these nodes may also be multiplied by the enhancements provided by DB2 Symmetric Multiprocessing for OS/400 (SMP). For example, a query that had previously been run on a single processor and is now being run on three four-way systems could experience a run time that is one-twelfth of what it had been. Although this amount of improvement may not be realized in most queries, there will still be many queries that will experience large improvements in performance.

- Queries that read and process a small number of records may experience some level of performance improvement when running over distributed data, but the percentage of improvement will in many cases be much less than queries over large files. For queries of this type, the amount of improvement will often be a factor of the speed of the connection between the systems, and in some cases, this may cause the query to run longer than it had on a single system.

- Queries that read and process a large number of records but that also return a large number of records to the coordinating system will in many instances not experience the almost linear improvements mentioned above. In this case, the individual nodes may still be able to process a subset of the records efficiently, but the response time may be affected by how quickly the records in the individual answer sets can be transferred back to the coordinator and how quickly this system can receive and process them as well.

- The performance of join queries on distributed data is closely linked to how much data needs to be transferred between nodes to perform the join. The best performing join queries are where all of the corresponding records of the files being joined exist on the same node so that no data is moved to other nodes to perform the join. These types of joins should improve nearly linearly with the number of nodes, although this again depends on the amount of data that needs to be transferred back to the coordinating system and the additional processing that will be needed there. Other join operations that need to move data between nodes to do the join will vary widely in how much improvement is achieved, and in some cases, may end up with a significant degradation. For this reason, partitioning of commonly joined files needs to be planned such that the most common join operations end up moving only smaller amounts or no data between nodes. For a more detailed discussion on distributed join

performance, refer to Chapter 6 of the previously mentioned DB2 Multisystem for OS/400 document.

- Queries that specify selection criteria on a single file may end up doing all the processing of that query on a single node if the optimizer determines that all the records matching the criteria exist on that node. In this case, the amount of performance improvement for this type of query will vary depending on how quickly the system at that node can process the query and return the results to the coordinating system. However, there are certain restrictions that a query must meet in order to be directed to a single particular node. More information on this type of query can be found in Chapter 6 of the DB2 Multisystem document.

- For most distributed queries and in particular for queries involving ordering of data, it is best to specify the ALWCPYDTA(*OPTIMIZE) parameter on the Open Query File (OPNQRYF) and Start SQL (STRSQL) commands and also on the Create SQLxxx (CRTSQLxxx) commands. This option allows the optimizer the most flexibility in choosing what method to use (an index or a sort) to order the records on each node.

- To achieve the fastest retrieval of data from a distributed file, you can issue the Override Database File (OVRDBF) command with the DSTDTA(*BUFFERED) parameter specified. For more information on this option, refer to Chapter 5 of the previously mentioned DB2 Multisystem for OS/400 document.

In addition to the above information, there are many other items to consider to understand distributed query optimization and how to obtain optimal performance levels when using DB2 Multisystem support. The following items (as well as many of the above) are covered in the above mentioned DB2 for Multisystem document.

- ORDER BY and GROUP BY operations

- Reusable and non-reusable ODPs

- Temporary result writers (new for DB2 Multisystem)

- Optimizer messages

- Changes to the Change Query Attributes (CHGQRYA) command

# Chapter 5. Communications Performance

There are many factors that affect iSeries performance in a communications environment. This section discusses some of the common factors and offers guidance on how to achieve the best possible performance. Much of the information in this section 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 workload and other performance workloads. In some cases, the actual performance data is included here to reinforce the performance claims and to demonstrate capacity characteristics. The NetPerf workload is defined at the end of this chapter.

**Communications Performance Highlights for V5R1:**

- Again in V5R1, there was an intentional effort to further improve the performance of the communications infrastructure, building upon the significant performance improvements that were already introduced in V4R4 and V4R5. In V5R1, the scalability was significantly improved. Having efficient scaling means that as throughput increases, CPU consumption increases roughly proportionally to throughput.

- V5R1 also added new function to support the SSL GSK APIs and the VPN/IPSec RC5 symmetric cypher. Also added was support to optionally offloading a portion of the SSL handshake processing to the 4758-23 Cryptographic Coprocessor. This handshake or key-processing function is very CPU-intensive and has the potential of offload up to 90% of the iSeries server CPU consumption for full-handshakes scenarios.

- V5R1 eliminated the need for TCPONLY(*YES) which previously was required to reduce CPU consumption for TCP transmissions. Now internal and automatic, TCP transmissions can maintain good performance even when APPC is running concurrently over the same line without this extra configuration parameter. In addition, now in V5R1, multiple 100 Mbps Ethernet adapters attached to one IOP can benefit from the higher performance level. Prior to V5R1, the higher level of performance was only available to IOPs that had a single Ethernet adapter.

- FTP performance and scalability has a higher potential in V5R1 by taking advantage of the new Asynchronous I/O Completion Ports sockets API interface.

**Highlights from V4R4 and V4R5:**

- V4R5 added support for Gigabit Ethernet on the PCI bus based Model 2xx and 8xx systems. The Gigabit Ethernet protocol supports a raw bandwidth of 10 times that of 100 Mbps Ethernet and over six times that of 155 Mbps ATM. Measurements on the iSeries demonstrate that it allows the majority of this high bandwidth to be utilized by real applications. Sustainable throughputs of over 700 Mbps have been achieved, that represents an increase of over 7 times that of 100 Mbps Ethernet. This extra throughput encourages high-bandwidth connections while simplifying the network topology.

- In V4R5, the model 2XX and 8XX systems include a higher-capacity native PCI bus. This provides a lower latency interface compared with the bus in previous models. Because of the decreased latencies, you may experience better throughput with 100 Mbps Ethernet compared with V4R4.

- The performance of encryption and security software improved significantly in V4R5 and as well as in V5R1.

- In V4R4, the performance and scalability of TCP/IP was significantly increased. Significant network infrastructure software performance enhancements were implemented including: Optimization of Sockets APIs and re-implementing Sockets and SSL in SLIC, optimization of TCP/IP to improve performance and scalability, and a more efficient scheme for the Ethernet device driver to interface with the IOP/IOA. These changes improved performance by significantly reducing the CPU time required by the communications software. The CPU time for the Request/Response scenario (client server like) was reduced by 40-50%. The CPU time for the Connect/Request/Response scenario (web server like) was reduced by 40-50%. The maximum transfer rate for the Streaming (large transfer) scenario using 100 Mbps Ethernet increased from 40 Mbps up to 90 Mbps. There were additional software optimizations in V4R5 and again in V5R1 to further reduce software contention. Measurements with n-way systems indicate significant capacity increases and better scalability compared with V4R3.

## 5.1 TCP/IP, Sockets, SSL, VPN, and FTP

### TCP/IP Capacity Planning and Performance Data

Table 5.1 provides some rough capacity planning information for communications when using Sockets with TCP/IP over 100 Mbps Ethernet. Variations with SSL and VPN are also included. This is based on measurements gathered from iSeries Model 2XX and 8XX systems. This table may be used to estimate a system's potential transaction rate at a given CPU utilization assuming a particular workload and security policy.

*Table 5.1. V5R1 iSeries TCP/IP Capacity Planning*

| | **Capacity Metric (transactions/second per CPW)** | | | | |
|---|---|---|---|---|---|
| **NetPerf Transaction Type:** | **Nonsecure TCP/IP** | **SSL (RC4 / MD5)** | **VPN (AH with MD5)** | **VPN (ESP with DES/MD5)** | **VPN (ESP with RC4/MD5)** |
| Request/Response (RR) 1 Byte | 16.3 | 10.4 | 7.4 | 5.1 | 5.9 |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 2.7 | 0.74 | 0.94 | 0.36 | 0.65 |
| Large Transfer (Stream) 16K Bytes | 10.7 | 1.3 | 1.2 | 0.28 | 0.70 |

**Notes**:

- Capacity metrics are provided for nonsecure and each variation of security policy
- Based on measurements with the NetPerf workload using various iSeries 2XX and 8XX models with V5R1
- The table data reflects iSeries as a server (not a client)
- The data reflects Sockets, TCP/IP and 100 Mbps Ethernet.
- Measurements used transport mode, 56-bit DES or RC4 with 128-bit key symmetric cypher and MD5 message digest with RSA public/private keys. VPN antireplay has been disabled.
- If any of these configuration characteristics are changed, performance may differ significantly.
- CPW is the "Relative System Performance Metric" from Appendix D. Note that the communications CPU capacities may not scale exactly by CPW.
- This is only a rough indicator for capacity planning. Actual results may differ significantly.

For example, if a user has a VPN connection supporting a small packet request/response application (Model 830/2402, 1-byte request/response, VPN/ESP with RC4/MD5) and wishes to use about 20% of the overall CPU for the network processing portion, then note the following calculation:

*4200 CPW * 20% * 5.9 transactions/second/CPW = 4956 transactions/second*

While it is always better to project the performance of an application from measurements based on that same application, it is not always possible. This calculation technique gives a relative estimate of performance. Notice also that it is based on Netperf, a primitive workload. This application does little more than issue calls to sockets APIs. This allows the user to understand the tradeoffs between the various communications scenarios. A real user application will have this type of processing as only a percentage of the overall workload. The Workload Estimator, described in Chapter 23, 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 te Workload Estimator may be valuable in some cases.

This information is of similar type to that provided in Chapter 6, Web Serving Performance. There are also capacity planning examples in that chapter.

Table 5.2 below illustrates CPU consumption instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 5.1. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

*Table 5.2. V5R1 iSeries SSL and VPN Relative CPU Time*

| NetPerf Transaction Type: | Relative CPU Time (Scaled to Nonsecure baseline for each transaction type) | | | | |
|---|---|---|---|---|---|
| | **Nonsecure TCP/IP** | **SSL (RC4/MD5)** | **VPN (AH with MD5)** | **VPN (ESP with DES/MD5)** | **VPN (ESP with RC4/MD5)** |
| Request/Response (RR) 1 Byte | 1.0 x | 1.6x | 2.2x | 3.2x | 2.8x |
| Asym. Connect/Request/Response (ACRR) 8K Bytes | 1.0 y | 3.7y | 2.9y | 7.6y | 4.2y |
| Large Transfer (Stream) 16K Bytes | 1.0 z | 8.1z | 9.0z | 38.3z | 15.3z |

**Notes**:

- Based on measurements with the NetPerf workload using various iSeries 2XX and 8XX models with V5R1
- The table data reflects iSeries as a server (not a client).
- The data reflects Sockets, TCP/IP and 100 Mbps Ethernet. Variation of the protocol may provide significantly different performance.
- Measurements used transport mode, 56-bit DES or RC4 with 128-bit key symmetric cipher and MD5 message digest with keyed RSA public/private keys. VPN antireplay has been disabled.
- This is only a rough indicator for capacity planning. CPU capacities do not scale exactly by CPW; therefore, actual results may differ significantly.
- x, y, and z are scaling constants, one for each NetPerf scenario.

Again, remember that this information is based on the NetPerf workload, which is a primitive workload. This application does nothing other than issue sockets APIs. A real user application will have this magnitude of CPU time for only a percentage of the total CPU time. Also the SSL and VPN measurements are based on specific set of cypher suites and public key sizes. Other choices will perform differently.

From Table 5.2, note the CPU Time required to process transactions in a secure mode. Some overheads are fixed while some are size related. The fixed overheads include the handshakes needed to establish a secure connection. 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.

Table 5.3 shows the relative throughputs of 100 Mbps Ethernet, Gigabit Ethernet and jumbo frame Gigabit Ethernet. These transfer rates were measured between two iSeries model 840s using the NetPerf workload. The TCP/IP send and receive buffers were set to 1 MB. The 16 KB Streaming scenario consists of a two way request/response. This sort of scenario might be part of an application doing large file copies over a network. The communications path is the focus of this scenario. A typical server application might also include other function/pathlength such as data encryption/decryption and database access.

*Table 5.3. V5R1 iSeries Gigabit Ethernet Performance*

| | **Aggregate Transfer Rate (Mbps)** | | |
|---|---|---|---|
| **NetPerf Transaction Type:** | 100 Mbps Ethernet 1496-Byte MTU | Gigabit Ethernet 1496-Byte MTU | Gigabit Ethernet 8996-Byte MTU (Jumbo Frame) |
| **Large Transfer (Stream) 16K Bytes** 1 user on 1 adapter | 93 | 737 | 985 |
| **Large Transfer (Stream) 16K Bytes** 4 users on 4 adapters | 366 | 2,145 | 2,827 |

## Performance Observations/Tips

- Gigabit Ethernet provides over 7 times more throughput than 100 Mbps Ethernet for a single user. If jumbo frames are used, then Gigabit Ethernet provides up to 10 times more throughput than 100 Mbps Ethernet for a single user.

- Gigabit jumbo frames allow higher throughput while decreasing system CPU pathlength. The jumbo frame MTU is 6 times larger than the standard frame MTU. This allows the per frame pathlength cost to be spread over 6 times as many bytes. NetPerf measured system capacity grew by approximately 30% when using jumbo frames rather than standard frames. This option requires jumbo frame or 8996 Byte MTU support by all of the gigabit network components including switches, routers and bridges. For configuration, LINESPEED(*AUTO) and DUPLEX(*FULL) or DUPLEX(*AUTO) must also be specified.

- 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. This means that there will be a one-to-one match between frames and MTUs.

- When transferring large amounts maximize the size of the application's send and receive size. This is the amount of data that the application transmits 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.

- Starting with V4R4, TCP/IP can take advantage of larger buffers. Prior to V4R4, the TCP/IP buffer size (TCPRCVBUF and TCPSNDBUF on the CHGTCPA or CFGTCP command) was recommended to be increased from 8K bytes to 64K bytes to maximize data rates. When transferring large amounts

of data with V4R4 , V4R5, or V5R1, you may receive 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. For Tables 5.1, 5.2 and 5.3, the 1MB buffer size was optimal.

- To receive the full benefit of the 100 Mbps Ethernet performance improvements in V4R4 and V4R5, it is essential that the **TCPONLY parameter in the LIND have a value of *YES and that the 100 Mbps Ethernet adapter have a dedicated IOP (**feature code 2842 or 2843 on the 2xx and 8xx models). This allows the IOP to have the TCP/IP optimized version of its microcode active. The IOP and device driver running on the iSeries CPU will then run in a TCP optimized mode. If other high-level protocols such as APPC are active on that line, then the **TCPONLY** parameter must be set to *NO for functional reasons. If Ethernet is used with TCPONLY(*NO) or if TRLAN is used or if the 100 Mbps Ethernet IOP is shared, then you will realize only a portion of the V4R4 and V4R5 performance improvements in terms of CPU time reduction. These restrictions were eliminated on V5R1, so that TCP transactions could always have the higher-level of performance.

- In V4R5 and V5R1 the minimum Request/Response round trip delay is less than 0.5 millisecond for TCP/IP using an Ethernet IOP (for V4R5, with TCPONLY=*YES) on a model 2XX or 8XX. The CPW value for the CPU, the size of Request/Response, along with the load on the system will impact the round trip delay time. This type of delay is most noticeable in user transactions that contain many individual communications I/Os (like database serving). Having a fast IOP is critical to response time for these client/server environments.

- 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.

- Applications should consider taking advantage of the new Asynchronous and Overlapped I/O Sockets interface. By using this interface in V5R1, FTP transfer rates improved from 50 Mbits/sec to 90Mbits/sec over a single 100 Mbps Ethernet connection. This interface allowed FTP to run in a multithreaded non-blocking mode. Additional implementation information is available in the V5R1 Sockets Programming guide. The Asynchronous and Overlapped I/O sockets interface is now also available on V4R5 via a PTF.

- Decide whether SSL or VPN provides the proper level of security for you. VPN works at the IP layer rather than the socket layer as with SSL. Hence, it is typically used to secure a broader class of data than SSL - all of the data flowing between two systems rather than, for example, just the data between two applications. Other important differences include SSL does not protect UDP data, SSL cannot automatically generate new encryption keys (dynamic VPN connection) and securing a connection using VPN is completely transparent to the application.

- SSL supports both full and cached handshakes. When a client makes a secure connection with SSL for the first time, handshake and certificate processing must occur. This is referred to as the *full SSL handshake*. Once this has been done, the client's information can stay in the server's session key cache. After that, until the cached entry expires, a *cached SSL handshake* may occur when the same client reconnects. Table 5.1 reflects regular SSL handshakes for the Connect/Request/Response scenario. A full SSL handshake can consume up to about 20 times more CPU than the cached SSL handshake.

The impact of the full handshake on CPU utilization can be minimized by using a 4748-23 Cryptographic Coprocessor. Offloading full handshake processing on to this coprocessor may save up to 90% of the full handshake host CPU path length.

- Use SSL functions and APIs wisely to minimize the number of secure transactions for a given application. Secure transactions require significantly more CPU time and will reduce overall transaction capacity.

- Connections and closes using Secure Sockets (SSL) are expensive. Limit the number of times that new SSL connections must be established. (i.e., leave the connection up if possible). Because of the handshake processing that must occur with each new connection, an SSL Connect/Request/Response uses three to four times more CPU than with a SSL Request/Response when the connection is already in place.

- Client authentication requested by the server is quite expensive in terms of CPU and should be requested only when needed. They use two to three times the CPU resource of full handshakes that only do server authentication.
.
- If possible, use RC4 rather than DES VPN encryption. This takes advantage of RC4 performance improvements over DES. Referring to tables 5.1 and 5.2, VPN(ESP with RC4/MD5) can have twice the capacity and half of the CPU time of VPN(ESP with DES/MD5).

- The performance of VPN will vary according to the level of security applied. In general, configure the lowest level of security demanded by your application. In many cases data only needs to be authenticated. Refer to Tables 5.1 and 5.2. While VPN-ESP can perform authentication, AH-only affects system performance significantly less than ESP with authentication and encryption. Another advantage of using AH-only is that AH authenticates the entire datagram, ESP, on the other hand, does not authenticate the leading IP header or any other information that comes before the ESP header. Packets that fail authentication are discarded and are never delivered to upper layers. This greatly reduces the chances of successful denial of service attacks.

- The iSeries supports a set of APIs for SSL enabling socket applications called "SSL APIs". V5R1 added support for GSK APIs. This allows you another way to securely enable an application for SSL. SSL and GSK performance is similar. The choice of using GSK APIs vs. SSL APIs, in most cases, will not impact performance. Generally speaking, the SSL handshake, SSL read and SSL write processing performance is very similar when using either the GSK APIs or SSL APIs. A difference may occur in that GSK APIs require more API calls to set up the secure environment and these additional calls may cause additional performance load on the machine. GSK APIs do have considerable more flexibility and capabilities for adjusting different SSL environmental values than SSL APIs.

- 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.

## 5.2 Cryptographic Coprocessor Performance

Section 5.2 provides performance information for iSeries running with the Cryptographic Coprocessor feature code number 4758-023. The Cryptographic Coprocessor offloads portions of cryptographic processing from the host CPU. The host CPU issues requests to the coprocessor. The coprocessor then executes the cryptographic function and returns the results to the host CPU. Because the Cryptographic Coprocessor handles selected compute-intensive functions, the host CPU is available to process other system activity.

The 4758 Cryptographic Coprocessor can be used in two ways. First, OS/400 SSL (secure sockets layer) network communications can use the 4758 Coprocessor to dramatically offload cryptographic processing related to establishing an SSL session (i.e., handshake). Table 5.4 below reflects this sort of usage. Secondly, custom applications can be written to the CCA (Common Cryptographic Architecture) APIs to access the crypto services of the Coprocessor. Typically, banking and financial applications use the Coprocessor in this fashion. Tables 5.5 through 5.8 below show performance in these applications.

Cryptographic performance is an important aspect of capacity planning, especially for applications using secure network communications. Besides host processing capacity reflected by the CPW rating, the impact of one or more Cryptographic Coprocessors must be considered. The information in this paper may be used to assist in capacity planning for this complex environment.

The data presented in this paper is not representative of a specific customer environment. Results in other environments may vary significantly. These measurements were completed on iSeries 8xx models, but the relative performance and recommendations are similar for other models.

**Workload Descriptions**

A variety of workloads were used that utilize the cryptographic ability of the Cryptographic Coprocessor.

- NetPerf with the ACRR scenario (see workload description in section 5.6) using SSL
  - This scenario includes full, rather than abbreviated, connect handshakes. This reflects the sort of CPU overhead experienced when a user begins a secure transaction. As implemented in V5R1, data authentication/encryption/decryption is handled by system CPU.
  - Typical connects today use RSA 1024 bit public/private key pairs. This scenario does the same.
  - This scenario used the SSL programming APIs. For a description of SSL APIs see "Secure Sockets Layer (SSL) APIs" -- http://publib.boulder.ibm.com/html/as400/v5r1/ic2924/info/apis/unix9.htm .

- CCA Application Test cases
  - These test cases interface directly to the Cryptographic Coprocessor using the CCA interface. They reflect the sort of performance to be expected by a user running custom programs for the Cryptographic Coprocessor.
  - For details concerning the programming interface used by these test cases see the CCA Basic Services Guide at: http://www.ibm.com/security/cryptocards.

**Measurement Results**

The web-like measurements in Table 5.4 were made between similarly configured iSeries model 840 systems over a dedicated Gigabit Ethernet LAN

*Table 5.4 - Cryptographic Capacity Planning*

**Full Handshake Capacity**
**4758-023 Cryptographic Coprocessor**
**iSeries V5R1 Model 840**

| Number of Cryptographic Coprocessors | Number Jobs | Full Handshake Throughput (Transactions/Second) | Server Capacity Metric (Trans/sec per CPW) |
|---|---|---|---|
| | | | |
| No Coprocessor Offload | 1 | 23.3 | 0.046 |
| No Coprocessor Offload | Multiple | 383 | 0.052 |
| | | | |
| 1 | 1 | 33 | 0.550 |
| 1 | Multiple | 134 | 0.744 |
| | | | |
| 8 | Multiple | 1017 | 0.742 |

**Notes**:
- In the case of coprocessor offload and multiple jobs, enough jobs (up to 134 per coprocessor) were run to fully utilize the Cryptographic Coprocessor
- Measurements included RSA (1024 bit key and using CRT), MD5 and RC4 (128 bit key)
- Only RSA connection handshaking is offloaded to the Cryptographic Coprocessor . MD5 hashing and RC4 encryption/decryption is always done in the host CPU.
- Server Authentication only

Consider a user who, for example, has a Model 830-2402 without the Cryptographic Coprocessor. This user might wish to service full handshake connections using about 20% of the overall CPU for network processing. According to Appendix D: "CPW, CIW and MCU Values for iSeries" in the iSeries Performance Capabilities Reference, the CPW rating for this particular model is 4200. Note the following calculation using the Server Capacity Metric from the table above:

*4200 CPW * 20% * .052 transactions/second per CPW = 43 connections/second*

If the same user installed Cryptographic Coprocessors and wishes to operate under the same conditions then:

*4200 CPW * 20% * .742 transactions/second per CPW = 623 connections/second*

Note that this level of throughput would require (623/134) five Cryptographic Coprocessors to handle the load.

Another example of using this same information is to consider a user with a Model 830-2403 (CPW = 7350). This user might expect approximately 200 new web-like connections per second and wishes to understand how these connections might impact other work on that same system.

Using the server capacity metric in Table 5.4:

*200 connections/second / .052 transactions/second per CPW = 3846 CPW*

Notice that 3846 CPW is 52% of the total system CPW rating. Out of the total system capacity, 52% will be consumed servicing the secure LAN connection. This leaves less than half of the system to execute logic associated with the transaction and/or to support other applications.

Similarly, if Cryptographic Coprocessors are installed:

*200 connections/second / .744 transactions/second per CPW = 269 CPW*

Only 269 CPW will be required to support the expected 200 connections/second transaction rate. Now only 3.7% instead of 52% of the system capacity will be consumed servicing full handshake connections. This would allow over 90% of the CPU to be allocated for applications.

While it is always better to project the performance of an application from measurements based on that same application, it is not always possible. The calculation technique above gives a relative estimate of performance. Notice also that it is based on a primitive workload. This application does little more than issue calls to sockets and secure sockets APIs. This allows the user to understand the tradeoffs between the various communications and security scenarios. While this does include all SSL and communications processing, a real user application will have this type of processing as only a percentage of the overall workload. The Workload Estimator, described in Chapter 23, 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

The Cryptographic Coprocessor measurements in the four following tables were made using a 4758-023 Cryptographic Coprocessor installed in a dedicated iSeries Model 820. Cryptographic Coprocessor test cases using the CCA interface measured raw coprocessor throughput for a variety of cryptographic functions.

*Table 5.5*

**Private Key Decrypt Throughput**
**4758-023Cryptographic Coprocessor**

| Number of Threads | Key Form | RSA Key Length (Bits) | Throughput (Transactions/Second) |
|---|---|---|---|
| 1 | Exponent/Modulus | 1024 | 64.1 |
| 10 | Exponent/Modulus | 1024 | 106.4 |
| | | | |
| 1 | Chinese Remainder Theorem | 1024 | 86.2 |
| 10 | Chinese Remainder Theorem | 1024 | 137 |

*Table 5.6*

**Symmetric Key Encrypt Throughput**
**4758-023 Cryptographic Coprocessor**

| Number of Threads | Encryption Algorithm | Data Length (Bytes) | Throughput (K Bytes/Second) |
|---|---|---|---|
| 1 | DES | 102400 | 11636 |
| 10 | DES | 102400 | 15515 |
| | | | |
| 1 | Triple DES | 102400 | 9143 |
| 10 | Triple DES | 102400 | 11636 |

*Table 5.7*

**PIN Throughput**
**4758-023 Cryptographic Coprocessor**

| Number of Threads | Total Repetitions | Total Time (seconds) | Throughput (PINs/second) |
|---|---|---|---|
| 1 | 5000 | 40.2 | 124 |
| 3 | 15000 | 116.3 | 129 |

*Table 5.8*

**Signing Performance**
**4758-023 Cryptographic Coprocessor**

| Number or Threads | RSA Key Form | RSA Key Length (Bits) | Throughput (Transactions/Second) |
|---|---|---|---|
| 1 | Exponent/Modulus | 512 | 116 |
| 1 | Exponent/Modulus | 1024 | 65 |
| | | | |
| 1 | Chinese Remainder Theorem | 512 | 128 |
| 1 | Chinese Remainder Theorem | 1024 | 88 |
| 1 | Chinese Remainder Theorem | 2048 | 37 |
| | | | |
| 10 | Exponent/Modulus | 512 | 152 |
| 10 | Exponent/Modulus | 1024 | 108 |
| | | | |
| 10 | Chinese Remainder Theorem | 512 | 149 |
| 10 | Chinese Remainder Theorem | 1024 | 133 |
| 10 | Chinese Remainder Theorem | 2048 | 99 |

**Notes:**
- Exponent of 65537 used
- Signing does not include hashing overhead

**Observations, Tips and Recommendations:**

- The Cryptographic Coprocessor is highly scaleable within the iSeries environment. One coprocessor supports 134 full SSL handshakes per second while eight support 1017. This measured throughput using 8 adapters is within 5% of ideal scaling.

- Up to eight Cryptographic Coprocessors are supported per system. These are highly scaleable in terms of host capacity and CPU requirements. From one to eight coprocessors, the Server Capacity Metric varied only from 0.744 to 0.742.

- By comparing Capacity Metrics with and without the Cryptographic Coprocessor it can be seen that this coprocessor offloads 93% of the host full SSL handshake CPU requirements.

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

**Additional Information and Contacts**

Extensive information about using iSeries V5R1 Cryptographic functions may be found under "Security" at the iSeries Information Center web site at: http://www.ibm.com/eserver/iseries/infocenter .

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

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 is available at: http://www.ibm.com/security.

iSeries Security White Paper, "Security is fundamental to the success of doing e-business" is available at: http://www-3.ibm.com/security/library/wp_secfund.shtml . Other V5R1 performance information can be found in the V5R1 iSeries Performance Capabilities Reference on line at: http://publib.boulder.ibm.com/pubs/pdfs/as400/V4R5PDF/as4ppcp4.pdf or at the iSeries Performance Management website at: http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm**.**

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

---

## 5.3 APPC, ICF, CPI-C, and Anynet

**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.

- In general TCP/IP provides better performance with V4R4. 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 server iSeries system sending each record repetitively using the default blocking provided by OS/400 to the client iSeries system provides the best level of performance.

- A large transfer with the server iSeries system flushing the communications buffer after each record (FRCDTA keyword for ICF) to the client iSeries system 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 server iSeries system sending each record requiring a synchronous confirm (e.g., CONFIRM keyword for ICF) to the client iSeries system 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 trade-off 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 AS/400s. 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 iSeries 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 linespeeds less than a given threshold. In some configurations, it will compress data when using LAN; this significantly slows down LAN transfers.

---

## 5.4 LAN and WAN

**LAN Media and IOP:**

- 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 CPU 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 any other supported WAN protocol. This is due to the desirable combination of having a high media speed along with optimized protocol software.

- When several sessions use a line or a LAN concurrently, the aggregate data rate may be higher. This is due to the inherent inefficiency of a single session in using the high-speed link.

- In order to achieve good performance in a multi-user interactive LAN environment it is recommended to manage the number of active users so that LAN media utilization does not exceed 50% for TRLAN or 25% for Ethernet environments with multiple users because of media collisions resulting in thrashing. Operating at higher utilizations may cause poor response time due to excess queuing time for the line. In a large transfer environment where there is a small number of users contending for the line, at any given time a higher line utilization may still offer acceptable performance.

- There are several parameters in the line description and the controller description that play an important performance role.

    - **MAXFRAME** on the line description (LIND) and the controller description (CTLD): Maximizing the frame size in a LAN environment is very important and supplies best performance for large transfers. Having configured a large frame size does not negatively impact performance for small transfers. Note that both the iSeries system 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. Note that the maximum frame size allowed is 16393 for TRLAN and that a smaller value is the default.

    - **TCPONLY** on the line description (LIND): The parameter activates a higher-performance software feature which optimizes the way in which the IOP and the CPU pass data. This can be set to a value of *YES if TCP/IP is the only protocol to be used (e.g., not APPC).

- When configuring an iSeries system with communications lines and LANs it is important not to overload an IOP to avoid a possible system performance bottleneck.

- For interactive environments it is recommended not to exceed 60% utilization on a LAN IOP. Exceeding this threshold in a large transfer environment or with a small number of concurrent users may still offer acceptable performance. Use the iSeries performance tools to measure utilization.

- Optimally configured, the Gigabit Ethernet adapter can have an aggregate transfer rate of over 800 Mbps. See Table 5.3.

- Optimally configured, the 100 Mbps Ethernet IOP/IOA can have an aggregate transfer rate of up to 50 Mbps for TCPONLY(*NO) and up to 90 Mbps for TCPONLY(*YES). Multiple concurrent large transfers may be required to drive the IOP at that rate. (This assumes the use of the most recent IOP).

- Similarly in a web server environment using 100 Mbps Ethernet, the IOP capacity may be up to 120 hits/sec for TCPONLY(*NO) and 245 hits/sec for TCPONLY(*YES). This assumes nonsecure transactions and static pages of about 10K bytes each.

- The TRLAN IOP can support aggregate transfer rates of almost 16 Mbps, which is media speed.

- It is especially important to have a high-capacity IOP available for file serving, data base serving, web serving or for environments that have many communications I/Os per transaction. This characteristic will also minimize the overall response time.

- Higher-performing TRLAN IOP/IOAs have the potential to overrun lesser capacity TRLAN IOP/IOAs. Many re-transmissions and time-out conditions exist here. Check the iSeries performance tools for these statistics. For APPC, this can be minimized or avoided by limiting the LANACKFRQ

and LANMAXOUT parameters to 1 and 2, respectively, which are the default values.

- A given model of the iSeries system can attach multiple IOPs up to a given maximum number. It is important to distribute the workload across several IOPs if the performance capability of a single IOP is exceeded. There are also some limitations on the number of stations that can be configured through a single LAN connection.

- The larger maximum frame size gives 16Mbit Token Ring emulation over ATM the advantage vs. Ethernet emulation over ATM.

**WAN Line and IOP**:

- Typically WAN refers to communications lines running at 64Kbps or slower. In recent years, other WAN types (like Frame Relay) have increased media speed up to several Mbps.

- In many cases, the communications line is the largest contributor to overall response time. Therefore, it is important to closely plan and manage its performance. In general, having the appropriate line speed is the most key consideration for having best performance.

- A common misconception exists in sizing systems with communications lines. It is incorrect to believe that each attached line consumes CPU resource in a uniform fashion, and therefore, exact statements can be made about the number of lines that any given iSeries model can support. For example, if the sales pages say that a particular iSeries model supports 64 lines, it does not mean that any given customer can run their workload fully utilizing those 64 lines. It is merely a rough guideline stating the suggested maximum for that model (in some cases, it is the maximum configuration possible).

- Communications applications consume CPU and IOP resource (to process data, to support disk I/O, etc.) and communications line resource (to send and receive data or display I/O). 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). Communications IOP resource is also consumed to support the line activity.

  So the best question to ask is NOT "How many lines does my system support?", but rather, "How many lines does my workload require, and what iSeries model is required to accommodate this load?".

- To estimate the utilization of a half duplex line:
  utilization = (bytes in + bytes out) * 800 / time / linespeed
  where time = total # of seconds
  and linespeed = the speed of the line in bits per second

- For a full duplex line (e.g., X.25, ISDN), the iSeries Performance Tools report utilization as follows:
  Utilization = (bytes in + bytes out) * 400 / time / linespeed

  For example, if the send direction is 100% busy and the receive direction is 0% busy, the Performance Tools will report an overall 50% line utilization.

- The system usually can drive the line to a high utilization for applications that transfer a large amount of data. The difference of the data rate and the line speed is due to the overhead of header bytes, line turn around 'dead' time, and application serialization.

- 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.

- For interactive environments, keeping line utilization below 30% is recommended to maintain predictable and consistent response times. Exceeding 50% line utilization will usually cause unacceptable response times. The line utilization can be measured with the iSeries performance tools.

- For large transfer environments, or for environments where only a small number of users are sharing a line, having a higher line utilization may yield acceptable response times. In fact, maximizing line utilization means maximizing throughput for that single job.

- For large transfers, use large frame sizes for best performance. Fewer frames make more efficient use of the CPU, the IOP, and the communications line (higher effective data rate).

- To take advantage of these large frame sizes, they must be configured correctly. The MAXFRAME parameter on the LIND must reflect the maximum value. For X.25, the DFTPKTSIZE and MAXFRAME must be increased to its maximum value. Also, go to the APPC and TCP sections to ensure other related parameters are optimized.

- Configuring a WAN line as full-duplex may provide a higher throughput for certain applications that can take advantage of that, or for multiple-user scenarios.

- In general, the physical interface does not noticeably affect performance for a given protocol assuming that all other factors are held constant (e.g., equal line speeds). For example, if SDLC is used with a line speed of 19.2 kbps, it would not matter if a V.35, RS232, or an X.21 interface was used (all other factors held constant).

- For SDLC environments, polling is an important consideration. Parameters can be adjusted to change the rate at which a line is polled. Polls consist of small frames sent across the line and are processed by the IOPs. Therefore, polling contributes to line utilization and IOP utilization.

- The CPU usage (i.e., CPU time per unit of data) for SDLC and X.25 is similar. Depending on the application design, BSC and Async may require more CPU.

- The CPU usage for high speed WAN 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 (e.g., 1-2 Mbps), 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

- The WAN-capable IOPs handle the load with a relatively low IOP utilization and generally won't be the system performance capacity bottleneck.. However, you may check the IOP's utilization by using the Performance Monitor.

- For interactive environments it is recommended not to exceed 60% utilization on the communications IOP. Exceeding this threshold in a large transfer environment or with a small number of concurrent users may still offer acceptable performance. Use the iSeries performance tools to measure utilization.

- Even though an IOP can support certain configurations, a given iSeries model may not have enough system resource (for example, CPU processing capacity) to support the workload over the lines.

- In communications environments where errors are common, the use of smaller frame sizes may offer better performance by limiting the size of the re-transmissions. Having errors may also impact the number of communications lines that can run concurrently.

- The values for IOP utilization in SDLC environments do not necessarily increase consistently with the number of work stations or with the amount of workload. This is because an IOP can spend more time polling when the application is not using the line. Therefore, it is possible to see a relatively high IOP utilization at low throughput levels.

## 5.5 Work Station Connectivity

There are many ways to attach work stations (WS) to the iSeries via communications. Each type can have different overheads with the CPU or the media and can have other unique performance characteristics. The quantifications are based on measurements and analysis that occurred several releases ago, but the comparisons are likely still similar.

**Work Station Connectivity:**

- Interactive transactions include CPU processing for WS connection and application processing. If the application is "light" (similar to a commercial workload), then the performance impact of how WS are connected can be significant. Here, the percentage of CPU consumed to process screen I/O is greater. If the application is "complex" (significantly more CPU processing per transaction), then the performance impact of connectivity type is less significant and the percentage of CPU consumed to process screen I/O is less.

- Attaching WS through communications consumes more CPU than 5250 local WS support does. Keep in mind that the actual overhead may also vary significantly with changes in the data stream (number of I/Os, number of bytes, number of fields, and other screen I/O characteristics). For the following comparisons, we can compare the overall amount of CPU processing done for a "light" application (i.e., the amount to handle the screen I/O plus that to process the application). These comparisons assume that only this application is running on the system without any other workload consuming CPU. All comparisons are done with respect to the 5250 local WS baseline.

- WS attached with 5250 target-side DSPT, with remote work stations with the 5495 controller, or with CA/400 increase overall application CPU requirements for communications by about 10% and therefore reduce potential iSeries capacity by 10%.
- WS attached with TELNET increase overall application CPU requirements by about 25% and therefore reduce potential iSeries capacity by 20%. This is due to additional CPU processing per transaction for TCP/IP software and TELNET. Note that using the IBM Network Station as a work station uses this method of attachment. Note that this overhead varies greatly based on the screen characteristics. This impact can vary greatly due to the characteristics of the data stream (# bytes, # fields, etc.).
- WS attached with VT100/VT220 increase overall application CPU requirements by about 60% and therefore reduce potential iSeries capacity by about 40%. This is due to additional CPU processing for TCP/IP, TELNET, and data stream translation. This impact can vary greatly due to the characteristics of the data stream (# bytes, # fields, function keys, etc.).
- WS attached with 3270 Remote Attach, DHCF, NRF, or SPLS increase overall application CPU requirements by about 25% and therefore reduce potential capacity iSeries capacity by 20%. This is due to additional CPU processing for communications and data stream translation per transaction. This impact can vary greatly due to the characteristics of the data stream (# bytes, # fields, complexity, etc.).
- Web server based packages that allow 5250/HTML sessions significantly increase CPU requirements by several fold and therefore reduce potential iSeries capacity by several times. This is due to additional CPU processing for communications, the web server, and 5250 to HTML conversions.

- **Passing Through an iSeries to an application on another system** has several possibilities:

  - 5250 DSPT (source side) is the baseline here. This "front-end" system only has to support the WS attachment processing and communications support to the server system with no application processing being it is just a source side.
  - TELNET (source side) uses several times more CPU time (2-5 times more depending on the screen characteristics) than 5250 DSPT because of additional processing for TCP/IP communications and more processing in the TELNET application.

  ❖ APPN intermediate node routing (intermediate system passing transaction from source to target) has a similar CPU time to 5250 DSPT (source).

- **Twinaxial controllers provide better performance than ASCII controllers.** This is primarily due to the increased line speed. The conversion from ASCII to EBCDIC is performed in the ASCII controller, so it is not an impact to iSeries CPU time. ASCII response time should be similar to the response time for a remote work station configuration with a similar line speed.

- Keep the line utilization below 30% for best performance when interactive users are attached. This will maintain predictable and consistent response times. Exceeding 50-60% line utilization will usually cause unacceptable response times.

- **Mixed interactive users and batch**: When interactive users and large transfers are running on a communications line concurrently, consider the following to keep interactive performance acceptable:

1. Use APPN transmission priority to prioritize the interactive users' transfers over that of the large transfer. (this is the preferred choice, as it does not penalize the large transfer when there is no interactive traffic)
2. Change the RU size to a lower value for the large transfer. This optimizes interactive response time at the expense of large transfer performance (note that overall CPU time will increase also for the large transfer).
3. Reducing the pacing values for the large transfer will also slow it down, allowing the interactive users more windows for getting on the line.

---

## 5.6 NetPerf Workload Description

The NetPerf workload is a primitive-level function workload to explore communications performance. The NetPerf workload consists of C programs that run between a client iSeries and a server iSeries. Multiple instances NetPerf can be executed over multiple connections to increase the system load. The programs communicate with each other using sockets or SSL programming APIs.

Whereas most 'real' application programs will process data in some fashion, 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 database accesses).

To demonstrate communications performance in various different ways, several scenarios with NetPerf are analyzed. Each of these scenarios may be executed with regular nonsecure sockets or with secure SSL:

1. **Request/Response (RR)**: the client and server send a specified amount of data back and forth over a connection that remains active. This is similar to client/server application environments.
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, and the connection is closed. This is a web-like transaction.
3. **Large transfer (Stream):** the client repetitively sends a given amount of data to the server over a connection that remains active.

# Chapter 6. Web Server and Web Commerce Performance

This section discusses iSeries performance information in web serving and web commerce environments. Specific products that are covered here include: HTTP Server (powered by Apache), HTTP Server (original), WebSphere Commerce Suite, WebSphere Payment Manager, and Connect for iSeries.

**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
- 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) iSeries Web Server and Applications
- iSeries processor speed
- utilization of key iSeries resources (CPU, IOP, memory, disk)
- web server performance characteristics
- application (e.g., CGI, servlet) performance characteristics

The primary focus of this section will be to discuss the performance characteristics of the iSeries as a server in a web serving and web commerce environment, provide capacity planning information, and recommend actions to achieve high performance. Having a high-performance network infrastructure is important for web environments; please refer to Chapter 5, “Communications Performance” for related information and tuning tips.

**Comparing traditional communications to web-based transactions:** For commercial applications, data accesses across the Internet differ 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
- now, with the Internet, the network may not be well-understood

**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, along with other published benchmark data (from other sources) 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).

---

## 6.1 Web Serving with the HTTP Server

The HTTP Server for iSeries has been enhanced to support the popular Apache Server. HTTP Server (powered by Apache) and the HTTP Server (original) are used to denote which server is being referred to. Generically, they will be referred to as the HTTP Server.

**The typical high-level flow for web transactions:** the connection is made, the request is received and processed by the server, the response is sent to the browser, and the connection is ended. To understand this 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 a program that runs on a client work station that 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. Files and programs exist on the iSeries to support the various transaction types. Each of the transaction types used are quite simple, and will serve a "Hello World" response page back to the client. Each of the transactions can be served in a secure (HTTPS:) or a nonsecure (HTTP:) fashion.

- **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.
- **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. For more information on program activation groups refer to *iSeries ILE Concepts, SC41-5606*.
- **Persistent CGI:** HTTP invokes a CGI program which receives a handle supplied by the browser, and then builds a simple HTML page and serves it via the HTTP server.
- **Net.Data**: HTTP invokes a CGI program with a Net.Data macro that builds a simple HTML page and serves it via the HTTP server.
- **Servlet**: HTTP invokes a Java servlet which builds a simple HTML page and serves it via the HTTP server.
- **Apache User Module**: HTTP works in conjunction with an user-written module program to build a simple HTML page and serve the result.

**Web Server Capacity Planning**: Please use the iSeries Workload Estimator to do capacity planning for using static pages (the HTTP workload), WebSphere (the WebSphere workload), WebSphere Commerce Suite (the Web Commerce workload), and WebSphere Payment Manager (the Web Commerce workload). 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://as400service.ibm.com/estimator. Work with your marketing representative to utilize this tool (also see Appendix B).

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.

**Performance Metrics:**

- *"Capacity Metric: (transactions/second per CPW)"* is a relative indicator of server capacity using a particular type of transaction. It is derived from measurement: *(trans/sec) / (CPW value*CPU util)*.
- *"CPU Time Metric: (CPW per transaction/second)"* is a relative indictor of CPU consumption using a particular transaction. It is derived from measurement: *(CPW value*CPU util) / (trans/sec)*. Examples for using either metric are given in the conclusions.

*Table 6.1 V5R1 iSeries Web Serving Capacity Planning - Various Transactions*

| Transaction Type: | Nonsecure | | Secure | |
|---|---|---|---|---|
| | Capacity Metric: trans/sec per CPW | CPU Time Metric: CPW per trans/sec | Capacity Metric: trans/sec per CPW | CPU Time Metric: CPW per trans/sec |
| **Static Page (cached)** | 2.05 | 0.49 | 0.82 | 1.22 |
| **Static Page (not cached)** | 1.51 | 0.66 | 0.69 | 1.45 |
| **CGI (new activation)** | 0.08 | 12.10 | 0.08 | 12.50 |
| **CGI (named activation)** | 0.40 | 2.50 | 0.31 | 3.22 |
| **Persistent CGI** | 0.32 | 3.13 | 0.26 | 3.85 |
| **Net.Data** | 0.22 | 4.55 | 0.20 | 5.00 |
| **Servlet (WebSphere)** | 0.49 | 2.04 | 0.34 | 2.94 |
| **User Module** | 1.21 | 0.82 | 0.62 | 1.61 |

**Notes/Disclaimers**:
- IBM HTTP Server (powered by Apache) for iSeries; V5R1; WAS 3.5.3; 100 Mbps Ethernet
- Based on measurements from an iSeries Model 270, with a moderate web server load
- Data assumes no access logging, no name server interactions, no keepalive, LiveLocalCache off
- Secure: 128-bit RC4 symmetric cipher and MD5 message digest with 1024-bit RSA public/private keys
- CPWs are "Relative System Performance Metrics" listed in Appendix D
- Web server capacities may not necessarily scale exactly by CPW, actual results may differ significantly
- iSeries CPU features without an L2 cache will have lower web server capacities than the CPW value would indicate
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.

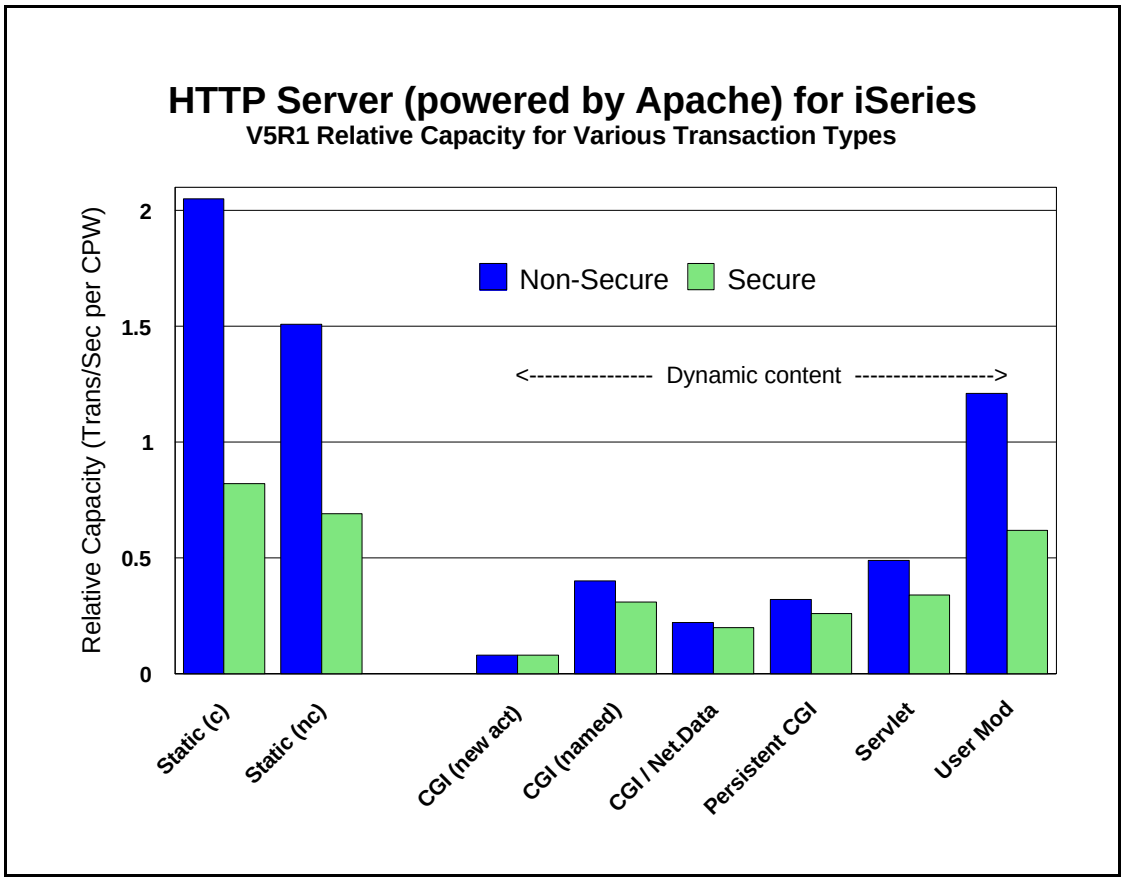



*Figure 6.1 iSeries Web Serving V5R1 Relative Capacities - Various Transactions*

*Table 6.2 V5R1 Relative Capacity for Static (varied sizes)*

| | Capacity Metric: transactions/second per CPW | | | |
|---|---|---|---|---|
| **Transaction Type:** | **1K Bytes** | **10K Bytes** | **100K Bytes** | **500K Bytes** |
| **Static Page (cached)** | 2.05 | 1.64 | 0.57 | 0.14 |
| **Static Page (not cached)** | 1.51 | 1.12 | 0.40 | 0.10 |

**Notes/Disclaimers**:
- IBM HTTP Server (powered by Apache) for iSeries; V5R1; 100Mbps Ethernet
- Based on measurements from an iSeries Model 270
- CPWs are "Relative System Performance Metrics" listed in Appendix D
- Web server capacities may not necessarily scale exactly by CPW, results may differ significantly
- iSeries CPU features without an L2 cache will have lower web server capacities than the CPW value would indicate

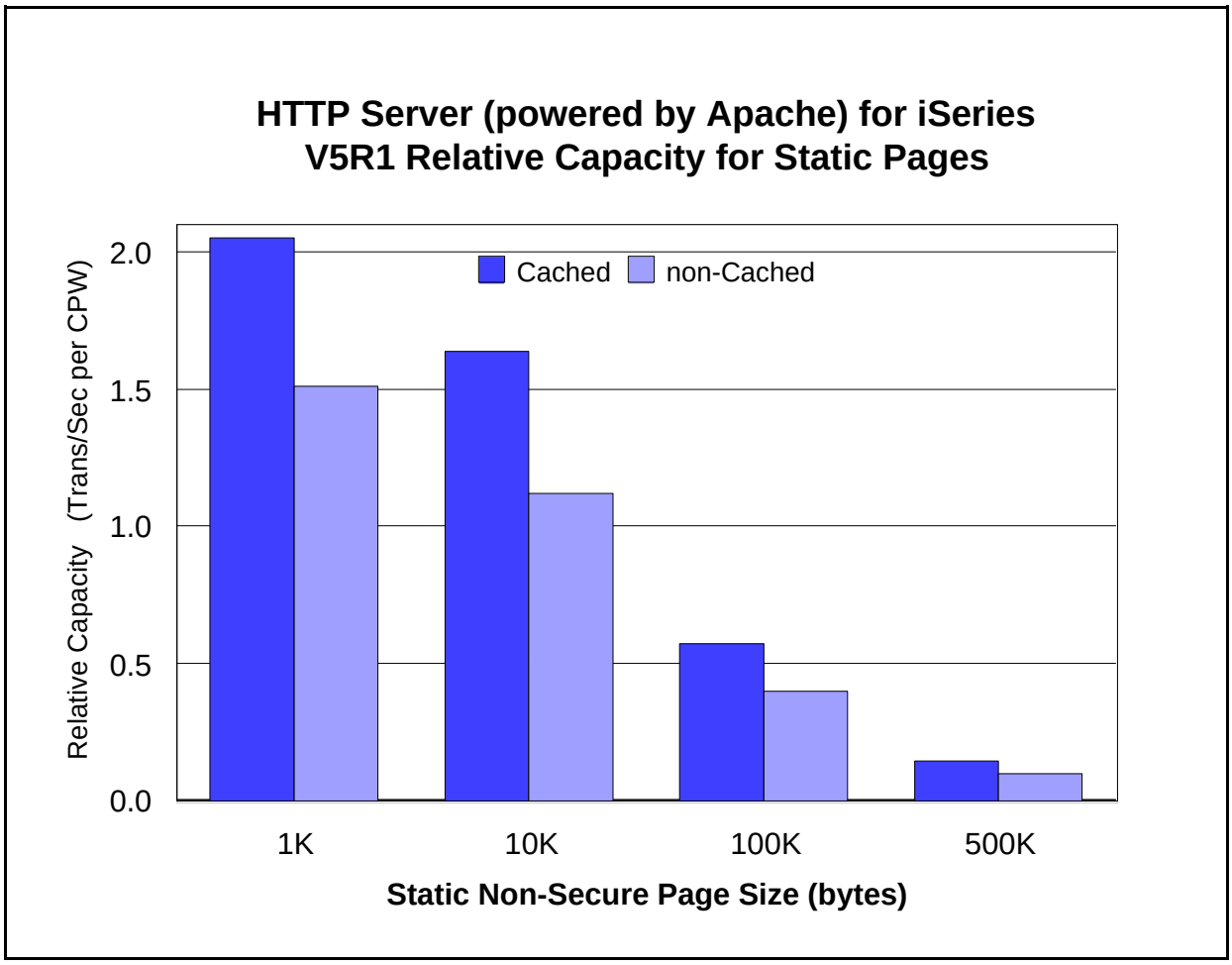



*Figure 6.2 V5R1 Relative Capacity for Static* Pages (varied sizes)

**Web Serving Performance Tips and Techniques**:

1. **HTTP software optimizations by release:**

    a. **V5R1** provides the new HTTP Server (powered by Apache) and continues to support the HTTP Server (original). The performance information provided in the tables represents HTTP Server (powered by Apache). In general, the performance of the two servers is fairly similar, so at a high level, the tables also reflect the performance of the HTTP Server (original). *In order to achieve the best possible performance, especially with the HTTP Server (powered by Apache), make sure that you get the latest PTFs: www.ibm.com/eserver/iseries/software/http*

    b. **V4R5** provides similar web server performance compared with V4R4 for most transactions (with similar hardware). The performance of secure web transactions did improve due to optimizations in encryption programs.

    c. **V4R4** provided a performance improvement of up to 70% over that of V4R3 (with similar hardware). This was mostly due to improvements in the IBM HTTP Server and TCP/IP performance. For static pages that are not cached, V4R4 provided up to 7% more capacity. For static pages that are cached, V4R4 provided up to 20% more capacity. For CGI and Net.Data transactions, V4R4 provided up to 70% more capacity.

    d. **V4R3** provided a performance improvement in capacity of up to 65% over that of V4R2 (with similar hardware). This was mostly due to the improved efficiency of the IBM HTTP Server over that of the ICS/400 from V4R2. For static pages that are not cached, V4R3 provided up to 20% more capacity. For static pages that are cached, V4R3 provided up to 65% more capacity.

2. **Web Serving Capacity (Example Calculations):** Throughput for web serving is typically discussed in terms of the number of hits/second or transactions/second. Typically, the CPU will be the limiting resource that determines overall server's capacity. If the IOPs become the resource that limits system throughput, then the number of IOPs supporting the load could be increased. For system configurations where the CPU is the limiting resource, Table 6.1 above can be used for capacity planning. Use these high-level estimates with caution. They do not take the place of a complete capacity planning session with actual measurements of your particular environment. Remember that these example transactions are fairly trivial. Actual customer transactions may be significantly more complex and therefore consume additional CPU resources. Scaling issues for the server, the application, and the database also may come into consideration when using N-way processors with relatively higher projected capacities.

    a. **Example 1: Estimating the capacity for a given model and transaction type:** Estimate the system capacity by multiplying the *CPW* (relative system performance metric) for the iSeries model with the appropriate *transactions/second per CPW* value (the capacity metric provided in Table 6.1). *Projected Capacity at 100% CPU = CPW * trans/sec/CPW* For example, a 270-2432 rated at 1070 CPWs doing web serving with noncached static pages, would have a capacity of 1615 trans/sec (1070 CPWs * 1.51 trans/sec/CPW = 1615 trans/sec). This assumes that the entire capacity of the system would be allocated to Web serving. If other work will also be on the system, you must pro-rate the CPU allocation. For example, if only 20% of the CPU is

allocated for Web serving, then it would have a web serving throughput of 323 trans/sec (1070 CPWs * 1.51 trans/sec/CPW * 20% = 323 trans/sec).

b. **Example 2: Estimating how many CPWs are required for a given web transaction load**: Characterize the transaction make-up of the estimated workload and the required transaction rate (in transactions/second). Estimate the CPWs required to support a given load by multiplying the required transaction rate by the appropriate *CPW per transactions/second* value (the CPU time metric provided in Table 6.1). *Required CPWs = transaction rate * CPW/trans/sec*. For example, in order to support 825 noncached static trans/sec, 545 CPWs would be required (825 trans/sec * 0.66 CPW/trans/sec = 545 CPWs) . If a mixed load is being assessed, then calculate the required CPWs for each of the components and add them up. Select an iSeries model that fits, having an acceptable resulting CPU utilization, and allows enough room for future growth.

3. **Web Server Cache for IFS Files**: Serving static pages that are cached can significantly increase web server capacity (refer to Table 6.2). Ensure that highly used files are selected to be in the cache. 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**.**

4. **Page size**: The data in the Table 6.1 assumes that a small amount of data is being served (say 100 bytes). Table 6.2 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. The IBM iSeries Workload Estimator can be used for capacity planning with page size variations (see Appendix B).

5. **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 iSeries ILE Concepts for more details on activation groups.

6. **Persistent CGI** is specific to applications needing to keep state information across web transactions. Don't confuse persistent CGI with persistent connections, they are totally different. Persistent CGI is not really a way to improve the performance of your CGI program, but more of a functional advantage. You'll notice in Table 6.1 that the performance of CGI is nearly identical to that of persistent CGI.

7. **Net.Data:** Net.Data macros run slower because the macro is interpreted (although macro caching improves performance) while the CGI program is compiled code. You should weigh the functional advantages in using Net.Data macros against the additional resources it consumes.

8. **Apache User Modules:** The HTTP Server (powered by Apache) provides support for user modules. These highly flexible user-written programs that are used in cases where you want the HTTP Server to pass control to you on each HTTP transaction. You can then choose to “decline” the transaction or process it with your user-written module. An implementation with user modules will generally provide higher server performance compared with more standard approaches (e.g., CGI, servlets, Net.Data, etc.).

9. **Secure Web Serving**: Secure web serving involves additional overhead to the server. Additional line flows occur (fixed overhead) and the data is encrypted (variable overhead proportional to the number of bytes). Note the capacity factors in the tables above comparing nonsecure and secure serving. For simple transactions (e.g., static page serving) the impact of secure serving is 2x, or perhaps more based on the number of bytes served. For complex transactions (e.g., CGI, Net.Data, servlets), the overhead is 40% or less.

10. **Persistent Requests and Keep Alive**: 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 Keep Alive. 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. 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.

11. **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.

12. **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.

13. **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 Requests or Keep Alive can improve this.

14. **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/pubs/html/as400/v4r5m1/ic2924/index.htm* and http://www.iseries.ibm.com/products/http/docs/v4r5/

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".

15. **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, using CGI programs and using Net.Data**.**

16. **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.

17. **Communications/LAN IOPs**: Since there are a dozen or more line flows per transaction, 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 Keep Alive, etc. For the best projection of IOP capacity, consider a measurement and observe the IOP utilization. The 2619 or the 2617 LAN IOPs have a capacity of roughly 70 trans/sec when serving small (e.g., 1K byte) nonsecure pages (keep in mind that each hit contains a dozen or so line flows). Using Ethernet or TRLAN IOPs from V4R1-V4R4, typically have capacities in the 100-130 trans/sec range. If 100M Ethernet is used and the TCPONLY parameter in the LIND has a value of *YES, then capacities of up to 250 trans/sec may be seen (V4R5 and earlier). For new V4R5 PCI bus IOPs, you might see capacities in the 500 trans/sec range.

## 6.2 WebSphere Commerce Suite

Use the iSeries Workload Estimator to predict the capacity characteristics for WebSphere Commerce Suite performance in business-to-consumer environment (using the Web Commerce workload category). The Workload Estimator will ask you to specific 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.). The Workload Estimator has been enhanced to also optionally allow WCS to be sized in conjunction with WebSphere Payment Manager to process payment transactions. You'll find the tool at: http://as400service.ibm.com/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 Appendix B).

Please refer to the WebSphere section in chapter 7, for a discussion on WebSphere Application Server performance as well as related web links.

## 6.3 WebSphere Payment Manager

Use the iSeries Workload Estimator to predict the capacities and resource requirements for WebSphere Payment Manager. The Estimator allows you to predict a standalone WPM environment or a WPM environment associated with the buy visits from a WebSphere Commerce Suite estimation. Work with your marketing representative to utilize this tool. The Workload Estimator is available at: http://as400service.ibm.com/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. Payment Manager 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 Payment Manager 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 chapter 7, for a discussion on WebSphere Application Server performance as well as related web links**.**

---

## 6.4 Connect for iSeries

IBM Connect for iSeries is a software solution designed to provide iSeries customers and business partners a way to communicate with an eMarketplace. Connect for iSeries was developed as a software integration framework that allows customers to integrate new and existing back-end business applications with those of their trading partners. It is built on industry standards such as Java, XML and MQ Series.
The framework supports plugins for multiple trading partner protocols. Connect for iSeries also provides pluggable connectors that make it easy to communicate to various back-end applications through a variety of access mechanisms. Please see the Connect for iSeries white paper located at the following URL for more information on Connect for iSeries.
http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

**"B2B New Order Request" Workload Description:** This workload is driven by a program that runs on a client work station that simulates multiple Web users. These simulated users send in cXML "New Order Request" transactions to the iSeries server by issuing an HTTP post which includes the cXML New Order Request file as the body of the message. Besides the Connect for iSeries product, other files and back-end application code exist to complete this transaction flow. For this workload, XML validation was disabled for both requests and response flows. The intention of this workload is to drive the server with a heavy load and to quantify the performance of Connect for iSeries.

**Measurement Results:** One of the main focal points was to evaluate and compare the differences between the back-end application connector types. The five connector types compared were the Java, JDBC, MQ Series, Data Queue, and PCML connectors. The graphs below illustrates the relative capacities for each of the connector types. Please visit this link to learn about differences in connector types.

http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

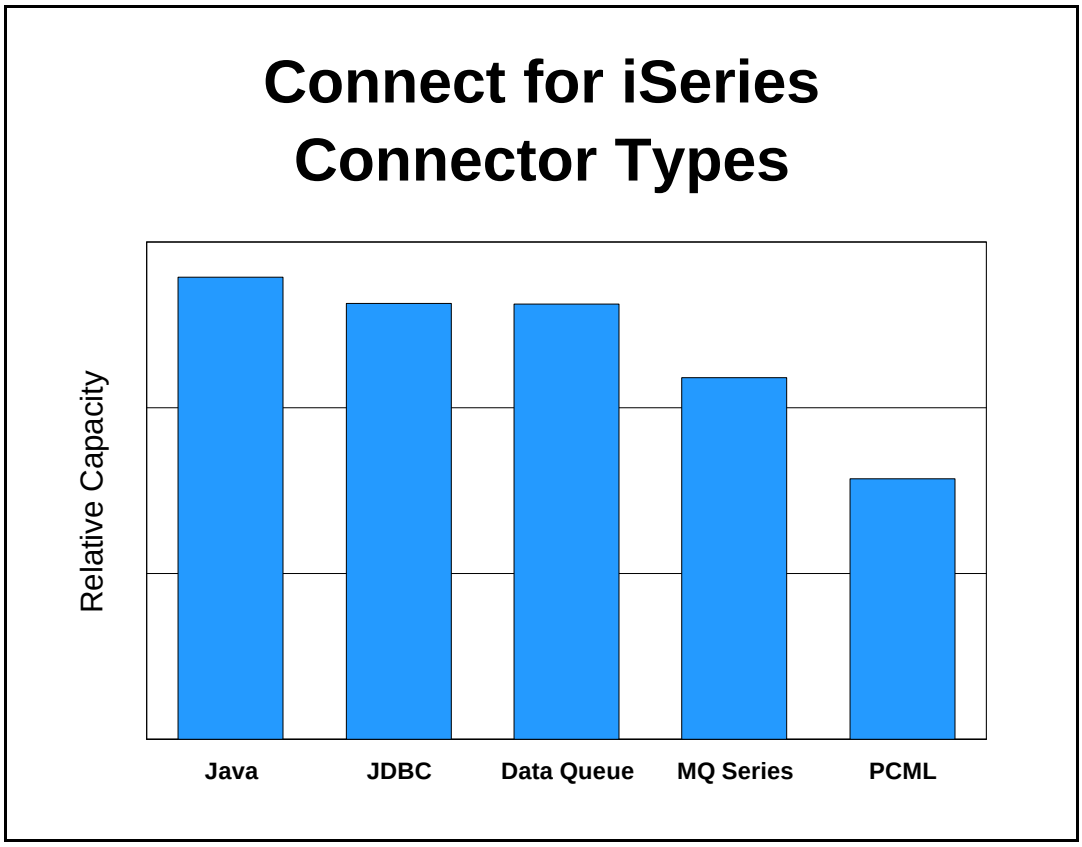



*Figure 6.3 Connect for iSeries - Connector Types*

**Performance Observations/Tips:**

1. **Connector relative capacity:** The different back-end connector types are meant to allow users a simple way to connect the Connect for iSeries product to their back-end application. Your choice in a connector type may be dictated by several factors. Clearly, one of these factors relate to your existing back-end application and the programming language it is written in. This, in itself, may limit your choice for a back-end connector type. Please see the Connect for iSeries white paper to assist you in understanding the different connector types.
   http://www-1.ibm.com/servers/eserver/iseries/btob/connect/pdf/whtpaperv11.pdf

   Performance was measured for a simple cXML New Order Request. The Java connector performance may vary depending on the code you write for it. All connectors "mapped" approximately the same number of "fields" to make a fair comparison. The PCML connector has overhead associated with it in starting a job for each transaction via "SBMJOB". You can pre-start a pool of these jobs which may increase performance for this connector type.

2. **XML Validation:** XML validation should be avoided when not needed. Although many businesses will decide to have this feature on (you may not be able to assume the request is both "well formed and

validated") there are significant performance implications with this property "on". One thought would be to enable XML validation during your testing phase. Once your confident that your trading partner is sending valid and well-formed XML, you may want to disable XML validation to improve performance.

3. **Tracing:** Try to avoid tracing when possible. If enabled, it will impact your performance. However, in some cases it is unavoidable (e.g. trouble shooting problems)**.**

4. **Management Central Logging:** This feature will log transaction data to be queried and viewed with Management Central. Performance is impacted with this feature "on" and must be taken into consideration when deciding to use this feature.

5. **MQ Series Management Central Audit Queue:** Due to the fact that the Management Central Auditing logs messages into a MQ Series queue for processing, the default queue size may not be large enough if you run at a very high transaction rate. This can be adjusted by issuing wrkmqm and selecting the queue manager for your Connect for iSeries instance, selecting option 18 (work with queues) on that queue manager, selecting option 2 (change) and increasing the Maximum Queue Depth property. This property, when enabled, added approximately 15% overhead to the "B2B New Order Request" workload.

6. **Recovery (Check pointing):** Enabling transaction recovery adds significant overhead. This should be avoided when not needed. This property when enabled added approximately 50% overhead to the "B2B New Order Request" workload.

7. **MQ Series Connector Queue Configuration:** By default, in MQ Series 5.2, the queue manager uses a single threaded listener which submits a job to handle each incoming connection request. This has performance implications also. The queue manager can be changed to having a multithreaded listener by adding the following property to the file \QIBM\UserData\mqm\qmgrs\QMANAGERNAME\qm.ini Channels:

   ThreadedListener=Yes

   The multithreaded listener can boast a higher throughput, but the single threaded listener is able to handle many more concurrent connections. Please see MQ Series site for help with MQ Series. http://www-4.ibm.com/software/ts/mqseries/messaging/

# Chapter 7. WebSphere and Java Performance

Highlights:

- Introduction
- Hardware Improvements in V5R1
- Just In Time Compilation in Java
- WebSphere Performance, References for Tips and Techniques
- Java Performance -- Tips and Techniques
- Capacity Planning

## 7.1 Introduction

In traditional OS/400 applications, the performance of the application program itself is often a small contributor to overall performance. A large percentage of the execution is system services (e.g. Database Get Records) used by the application. Two ways to improve application performance are: 1) IBM improving OS/400, 2) The customer improving how the application uses the system services (especially database) in OS/400.

For Java, this can still be true. Key portions of Java (such as JDBC, encryption, security) can have a substantial portion of their support executing in OS/400. For some applications, tuning Java's use of these system services is performance tuning enough. However, it is also true that Java, as part of its portability story, will often have a higher percentage of the application's execution in Java programs and use less of a given Operating Service's function. Increasingly, it is the performance in the context of these Java "middleware" functions that is becoming important.

Java is now maturing as a language. Up until now, it has been of great interest to compare Java to traditional languages. While such comparisons are always difficult, in V4R5, this document suggested that Java computation had occassionally reached parity for some and was seldom to never more than two times slower than traditional languages. In short, performance had, even by V4R5, ceased to be a significant barrier to Java deployment. V5R1 does not change that.

But, such comparisons are becoming less relevant even as Java made substantial progress over the last several releases. Java has become important in its own right. Products such as Connect for iSeries and the WebSphere Application Server simply require Java and other advanced function, like XML, will find Java the likeliest choice.

In fact, the world of the web, servlets, and the emerging e-business function in general is becoming the premier place to deploy Java on iSeries. Accordingly, tuning the WebSphere environment is becoming increasingly important. Many may find tuning WebSphere matters more, at this point, than further mastery of Java language performance.

This chapter, which formerly devoted itself entirely to Java, will now pay equal attention to WebSphere since it is in this context that Java will increasingly be seen on iSeries. Other uses of Java will continue, of course, but WebSphere is going to be a primarly location for Java code and a platform from which to

launch important Java functionality, such as Servlets, Java Server Pages (JSPs), and Enterprise Java Beans (EJBs).

## 7.2 Hardware Improvements

V5R1 introduces new hardware with performance improvements above those delivered in V4R5.

Generally, one would expect a proportionate increase in Java performance corresponding to this new hardware. The main caveat is that CPW ratings will sometimes overstate the difference between Java applications running on the same model (see “CIW versus CPW for Java” in the Capacity Planning section).

Of continued interest are certain feature codes first seen in the V4R5 270 line. Those considering machines whose work has minimal 5250 content (such as a machine dedicated to WebSphere or Java applications, where there is virtually no 5250 applications content) might particularly look at these new processor feature codes and these machines to improve price/performance.

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. JDBC, Java's primary database access technique, is one factor that pushes up costs. In addition, many Java applications have more function than a corresponding RPG or COBOL application would have. So, even as Java reaches parity in terms of language code generation, many application writers tend to ask it to do more work than would have been the case for the same application in RPG or COBOL. For instance, Java also tends to get involved more in networking and various forms of data transformations (e.g. XML) that RPG and COBOL don't participate in as strongly if at all. Thus, Java will continue to require more cycles to get its typical application done than the more traditional languages because it is required to do more and different things.

This means that some models, such as the 250 models, are not really intended for a typical, Java-heavy deployment. In the right circumstances, such as a lightly-loaded storefront walk-up, with only a handful of users doing simple things, and especially with a working prototype to suggest precise workload costs, Java could be suitable on these machines. In general, however, WebSphere and other Java environments will tend to produce applications requiring 270 or 820 class machines at a minimum. The Workload Estimator will suggest suitable models, based on input assumptions you provide.

## 7.3 Just In Time Compilation

Java on other platforms have long featured a technology called "Just in time" (JIT) compilation. OS/400 Java, by contrast, features the Transformer technology. The Transformer creates hidden, compiled programs associated with the .class or .jar file. The resulting programs are called Direct Execution programs that are fully compliant with the Java standards. Our Java Transformer usually gives better computational performance, at least at the highest optimization levels, but there can be important exceptions. Moreover, the hidden program is transparent except for performance; the Java class file

remains exactly as it was in IFS and can even be run interpretively, with suitable parameters on the Java command.

If a class has not been subjected to the Transformer, the Java command's defaults would, prior to V4R5, subject it to a default transformation at a particular optimization level and then use the transformed program for execution. (The CRTJVAPGM command can create transformed programs explicitly).

Starting in V4R5, such classes are no longer transformed by default. The JIT is invoked instead.

In very dynamic environments, where classes are dynamically loaded frequently and not executed for long periods of time, the Just In Time compilation feature can sometimes offer advantages. So, in V4R5, OS/400 Java added Just In Time compilation as one option for managing overall Java performance.

In addition, some middleware, such as WebSphere, relies heavily on the use of “user class loaders”. Due to the way these class loaders worked, classes loaded with them will not use a pre-created Java program object, and will therefore run using the JIT by default. While it is usually possible to configure the server so that it will run with Direct Execution, running with JIT will generally offer reasonable performance in these environments. However, it may be worthwhile to go through the extra configuration steps for large libraries of code.

The JIT is also an easier way to do some forms of performance analysis. For instance, a typical invocation of CRTJVAPGM will not include "Entry/Exit" hooks. This means that some important information will not be available for some uses of the Performance Explorer. For large Java jar files, or directories with many class files, recreating the transformed class files using CRTJVAPGM can be prohibitively costly in terms of recompile time. However, re-running the application using the JIT can easily make this added information available without altering the underlying Java program created by the transformer. (By specifying INTEPRET(*JIT) and PROP((os400.enbpfrcol)), the *PGM created by the transformer is ignored for the current invocation of the JAVA command). In some cases, the application will usually be slightly slower under the JIT, and thus might sometimes obscure the problem under study. In most cases, the problem instead will become clear due to the presence of the extra information in the reports.

As a rough guide, the JIT technology runs at about the equivalent performance to Optimize(30).

## 7.4 WebSphere Application Server Performance

WebSphere is available in several versions. Choosing which to deploy is one significant factor. Another is the tuning, which varies in detail a bit, for each version of WebSphere. To help demonstrate these factors, the Trade2 Benchmark will show some of the relevant factors.

### Trade2 Benchmark (WebSphere eBusiness Benchmark)

The WebSphere Performance team has built a benchmark for characterizing performance of the WebSphere Application Server called the **WebSphere eBusiness Benchmark.** The benchmark was derived from experiences with many customer environments. The WebSphere eBusiness Benchmark was built to emulate an online brokerage firm. ***Trade2***, which is the actual application name, is a versatile test case designed to measure aspects of scalability, performance and competitiveness. The Trade2 application

is a collection of Java classes, Java Servlets, Java Server Pages and Enterprise Java Beans which together form an application providing emulated brokerage services. Figure 6.3 shows the system topology in which the Trade2 application runs. Trade2 is a follow-on to the Trade and Broker benchmarks used in previous reports. Trade2 was developed using the VisualAge for Java and WebSphere Studio tools and each of the components are written to open web and Java Enterprise APIs making the Trade2 application portable across J2EE compliant application servers. Trade2 follows the "WebSphere Application Development Best Practices for Performance and Scalability".

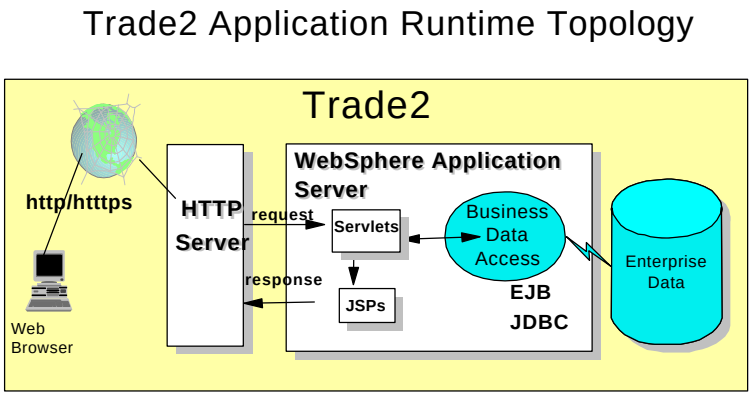



**Figure 7.1 Topology of theTrade Application**

The Trade2 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.

To measure performance across various configuration options, the Trade2 application can be run in several modes. A mode defines the environment and components used in a test and is configured by modifying settings in a profile. For example, data object access can be configured to use JDBC directly or to use

EJBs under WebSphere Advanced Edition by setting the Trade2 *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.

**Base Trade2 Benchmark Results**

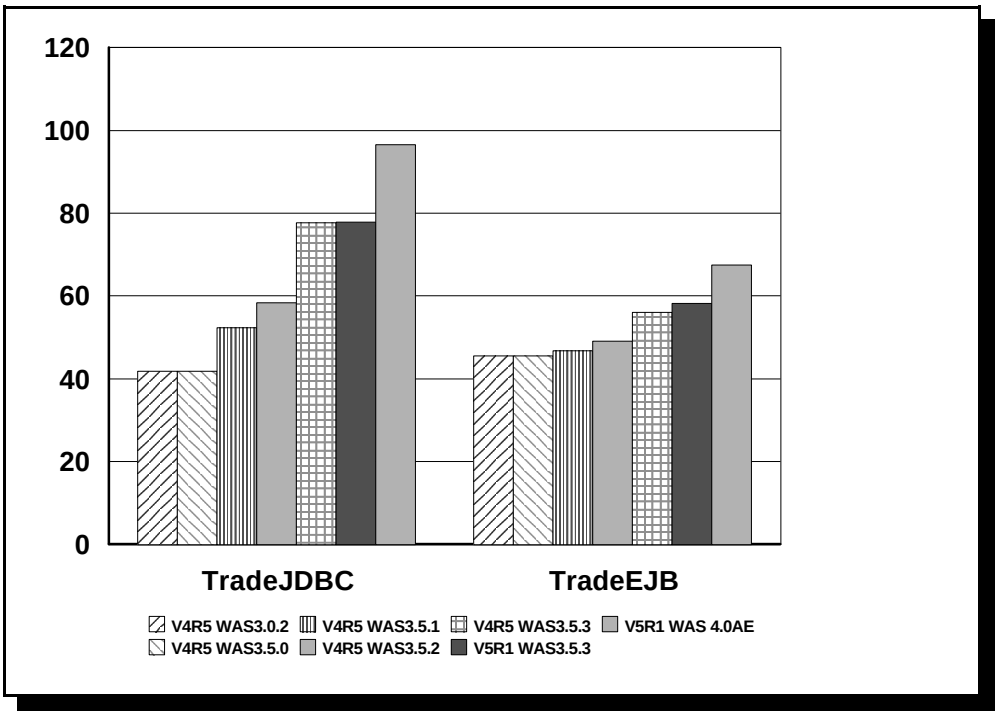



**Figure 7.2 Trade2 1-way Results**

| *WebSphere Application Server Trade2 Results* |
|---|
| **Notes/Disclaimers**: |
| • Results were measured on a 170/2385 system<br>• Trade2 JDBC and Trade2 EJB benchmarks<br>• WebSphere 3.0.2, 3.5.0, 3.5.1, and 3.5.2 were on a V4R5 system<br>• WebSphere 3.5.3 was measured on both V4R5 and V5R1<br>• WebSphere 4.0 AE was measured on V5R1<br>• IBM HTTP Server |

WebSphere Application Server Advanced Edition 3.5.3 provides significantly better throughput than older versions of WebSphere. Other measurements have shown that response time and scalability are also significantly improved with version 3.5.3.

### Trade2 Scalability Results

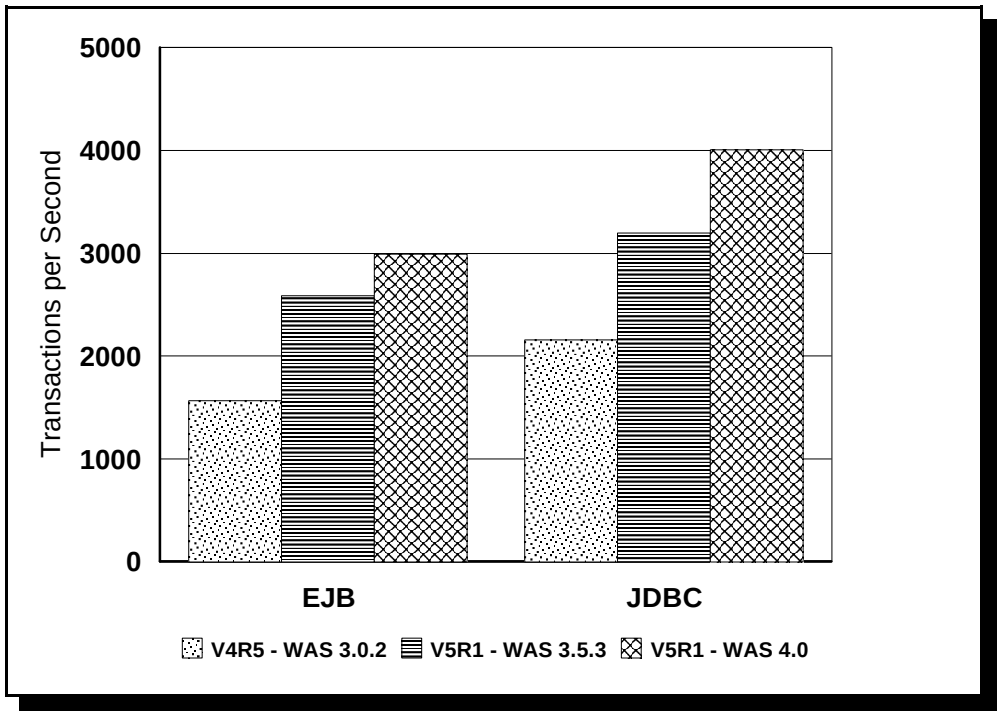



**Figure 7.3 Trade2 24-way Results**

| *WebSphere Application Server Trade2 Results* |
|---|
| **Notes/Disclaimers**: |
| • Results were measured on a 24-way system<br>• V4R5 - 840/2420<br>• V5R1 - 840/2461<br>• Trade2 JDBC and Trade2 EJB benchmarks<br>• V4R5 was measured with WebSphere 3.0.2<br>• V5R1 was measured with WebSphere 3.5.3 and WebSphere 4.0 AE<br>• IBM HTTP Server |

The new V5R1 24-way 840/2461 system with WebSphere Application Server 4.0 AE provides significantly better throughput than older versions of WebSphere running on the fastest 24-way.

### WebSphere Application Server 4.0 for iSeries Performance Considerations

The *WebSphere Application Server 4.0 for iSeries Performance Considerations* document describes the performance differences between WebSphere Application Server versions 4.0 and 3.5 on iSeries. It also contains many performance recommendations for environments using servlets, Java Server Pages (JSPs),

and Enterprise Java Beans. Some capacity planning information is included, though the Workload Estimator should be used for sizing WebSphere environments.

**WebSphere Application Server 3.0.2 for AS/400 Advanced Edition - Case Study for Improved EJB Performance**

The *WebSphere Application Server 3.0.2 for AS/400 Advanced Edition - Case Study for Improved EJB Performance* document shows tips and techniques to improve the performance of WebSphere EJB applications on AS/400 and iSeries. It contains specific examples of tools that can be used to find performance problems and tuning techniques for both WebSphere Application Server and iSeries.

To access these documents use the following URL:

**http://www-1.ibm.com/servers/eserver/iseries/software/websphere/wsappserver/product/PerformanceConsiderations.html**

**Workload Estimator**

See later in this chapter ("Capacity Planning") for both a link to the Workload Estimator and some suggestions on how to think about capacity planning generally.

---

## 7.5 Java Performance -- Tips and Techniques

**Introduction**

Tips and techniques for Java fall into several basic categories:

1. OS/400 Specific. These should be checked out first to ensure you are getting all you should be from your OS/400 Java application.

2. Java Language Specific. Coding tips that will ordinarily improve any Java application, or especially improve it on OS/400.

3. Database Specific. Use of database can invoke significant path length in OS/400. Invoking it efficiently can maximize the performance and value of a Java application.

4. Garbage Collection and Allocation Specific. Because Java programmers don't directly return their unused storage for reuse, the Java garbage collection facility must run on occasion to claim unused storage. Tuning the execution of garbage collection can be highly important to performance. This can be done by tuning garbage collection's performance or by avoiding the creation of new objects (see also language specific suggestions).

**OS/400 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. Information on the OS/400 JVM can be found at the http://www-1.ibm.com/servers/eserver/iseries/ebusiness/java/ Developer Kit for Java Web Site. Information on the OS/400 Toolbox for Java can be found at the http://www-1.ibm.com/servers/eserver/iseries/toolbox/ Toolbox Web Site.

- *Use CRTJVAPGM at Optimization level 40 on .class files*
  Java .class files should be converted into direct execution (machine instruction) Java program objects through the CRTJVAPGM command. CRTJVAPGM invokes the OS/400 Java Transformer. Use this command before running any performance critical Java programs. The Program object is permanent and will be reused once it is created. To see if the hidden program exists, use the DSPJVAPGM command. Optimization levels 10, 20, 30 and 40 are supported on the CRTJVAPGM command. After debug, in most cases, optimization level 40 should be used. Opt 40 ordinarily gives the best performance. Opt 20 or the new Just In Time mode might be a better choice for debugging. In a few cases, the Just In Time compiler might be faster for a deployed program, but the transformer at Opt 40 still wins in most cases.

- *Relative Performance (Optimization level)*:
  Results of specifying a given optimization level will vary by application. For computation and call intensive applications the relative gains can be dramatic. Here are the relative performance gains for a well-known artificial intelligence algorithm that features balanced computation and allocation:

| | Relative time (bigger is slower) |
|---|---|
| Optimization level 40 | 1.00 |
| Optimization level -- JIT (no transformer) | 1.19 |
| Optimization level 30 | 1.31 |
| Optimization level 20 | 2.14 |
| Optimization level 10 | 3.03 |
| Interpretive | 16.07 |

  Comparisons based on V4R5, JDK 1.1.8. Similar magnitudes would be observed on V4R4 and earlier (this difference has been pretty stable, and has been confirmed in similar magnitudes for a completely different program).

- *Use CRTJVAPGM Optimization Level 40 on .zip and .jar files*
  CRTJVAPGM at Optimization Level 40 should ordinarily be used on .zip and .jar files. The JAVA/RUNJVA command in V4R5 will, by default, use the Just In Time compilation for .jar and .zip that have no direct execution program. This may often be satisfactory as the time to use the transformer (via CRTJVAPGM) can be large for .jar or .zip files. On the other hand, some significant optimizations are only available on level 40 transformed .jar and .zip files, making the time to do the CRTJVAPGM well worth the time for heavily used applications. Care should also be taken. Check to see if some important use of the JAVA command (or also the equivalent RUNJVA command) explicitly invokes INTERPRET(*OPTIMIZE), the former default. In this case, the optimization level specified on the OPTIMIZE parameter will be used for all .jar and .zip files instead of the JIT. With very large .jar or .zip files, the time to implicitly transform a .jar or a .zip can look like a hang. To determine if your .zip/.jar file has a permanent, hidden program object, use the DSPJVAPGM command.

- *Upgrade existing .jar or .zip programs to Optimization Level 40 or delete the existing hidden program.*
  Because the JIT works at an equivalent of Optimize 30 or better, there is little point in having a .zip or .jar file with a transformed program at any lower optimization than optimization level 40. Either redo the CRTJVAPGM or do DLTJVAPGM to delete the hidden program. DLTJVAPGM does not affect the .jar or .zip file itself; only the hidden program.

- *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 starting in V4R4. Within a .zip or .class file, OS/400 Java will attempt to in-line code from other members of the .zip or .jar file.

- *Consider the special property* os400.defineClass.optLevel *for dynamically loaded .classes*
  Java's definition will occasionally cause the results of CRTJVAPGM to be ignored. This is especially true if your Java program loads a class "by hand" (Class.forname(), ClassLoader.loadClass()). In these cases, Java/400 cannot know the name of the file from which the class came, (strictly speaking, there may not be a file) so it must decide between interpretation and class loading using only the byte array provided by the defined interfaces. The os400.defineClass.optLevel property, which can be passed as a property through the Java command, will tell Java/400 whether to interpret or compile the program. Remember to pass the name and optimization level properly:

  JAVA CLASS(your.main.class) PROP((os400.defineClass.optLevel 40))

  In many cases, the "Just In Time" compilation will be optimal. Note: Special configuration may be necessary for WebSphere to use Direct Execution for executing your application code. Consult the WebSphere documentation for details. For V4R5 and above, it is generally better to let WebSphere use the JIT for application code.

- *Be aware of some automatic re-creation of "hidden" programs starting in V4R4.*
  Java/400, to improve performance, is changing the internal format of the hidden *PGM object created by CRTJVAPGM. All existing V4R3 *PGM objects will be recreated on their first use at the same optimization level as in V4R3 and become V4R4 or V4R5 objects unless someone uses CRTJVAPGM on the .class, .jar, or .zip file before the first use.

  If no action is taken, no harm is done; the recreation of the hidden program will commence. However, this change means the first use of a Java class in V4R4 or V4R5 that was unchanged from the V4R3 migration may appear to run more slowly. If you do the CRTJVAPGM yourself at a relatively benign time, this overhead should not affect production use of your machine even this one time. Doing the CRTJVAPGM by hand before use will be particularly beneficial for .zip and .jar files. It also means that if your program runs slowly in V4R4, try it again and see if the slowdown goes away. If it does, some class probably underwent compilation for migration. *Note:* This is strictly a performance issue. You *do not need* to recompile your .java source or make any other changes to your program because of this activity. The classes shipped with OS/400 JV1 are already at the V4R5 level.

**Java Language Performance Tips**

- *Minimize synchronized methods*
  Synchronized method calls take at least 10 times more processing than a non-synchronized method call.

Synchronized methods are only necessary if you have multiple threads sharing and modifying the same object. If the object **never** changes after it is created ("constructed" is the Java term for "created"), you don't need to synchronize any of its methods, even for multithreading.

**Note**: Dealing with synchronized methods mean understanding some important Java programming concepts.

- Some Java objects, notably String, do not permit data in the object to be modified after the object is constructed. For such objects, synchronized methods are never needed.

- Other objects, such as StringBuffer, allow the object to be modified after construction. All of its methods are synchronized.

- Many objects fit these two models. If a StringBuffer type object will be used by even one multithreaded application, all methods must be synchronized except its constructors. If you never use multithreading, then a StringBuffer type object requires no synchronization.

- But, consider object reuse when you decide. If some later application uses your object, and that new application is multithreaded, synchronization will be needed. This is why common Java objects like StringBuffer have synchronized methods.

- *Minimize object creation*
  Object creation can occur implicitly within the Java APIs that you use as well as within your program. Object creation and the resulting garbage collection can typically take 15% to 30% of a server transaction workload. To minimize this cost you can reuse an object's space implementing a "reset" method that reinitializes the local variables in the object. The code fragment

```
if (objx == null)
  objx = new x(some,creation,parameters);
else
  objx.reset(some,recreation,parameters);
```

  can provide significant performance improvements.

  Common causes of object creation that may not be obvious:

  - The I/O function readLine() creates a new String.

  - Invoking the substring() function of a String creates a new String.

  - The JDBC Result Set function getString() creates a String.

  - The StringTokenizer returns a String from many functions.

  - Passing a scalar int or long as an object will create an Integer or Long object.

- *Minimize the use of String objects*
  String objects in Java are immutable. This means that you can not change (append, etc.) to a string object without creating a new object. Object creation is expensive and can occur multiple times for

each String object you are using. To minimize the use of String objects you should use either StringBuffer or char[]. StringBuffer may also be a problem since the StringBuffer classes use synchronized method calls. An array of characters (char[]) can be used to simulate fixed length strings. This is recommended for applications which make heavy use of string data.

*Relative Performance*:
The following table shows the relative performance difference when using String, StringBuffer, or char[]. The test case concatenates two strings. For the char[] case. the concatenation reduces to simple array assignment, thus avoiding the creation of objects and the synchronization overhead associated with StringBuffer. In the following table an initial String was concatenated to the string "Wait". For the char[] case there were simply four char[] assignments for the characters 'W' 'a' 'i' 't'. This operation was repeated four times (for a total character size of 16 and 16 "setup" operations). The result was then turned into a String object.

| | Relative time (bigger is slower) |
|---|---|
| char[] ('W' 'a' 'i' 't' | 1 |
| StringBuffer ("Wait" and 'W' 'a' 'i' 't') | 2.8 - 5 |
| String ("Wait" and "W "a" "i" "t") | 11.3 - 46.2 |

Comparisons based on Optimization level 40, V4R5. JDK 1.1.8.

- *Leverage variable scoping*
  Java supports multiple techniques for accessing variables. One typical technique is to write an "accessor" method. Local variables and instance (per object) variables are the fastest.

  *Relative performance*:
  Here are five comparisons on variable access time and their relative performance. A local variable is the fastest and is given a relative performance of 1.

| | Relative time (bigger is slower) |
|---|---|
| Local variable | 1.0 |
| Instance variable:* | 1.0 |
| Accessor method in-lined: | 4.8 |
| Accessor method: | 4.8 |
| Synchronized accessor method: | 68.8 |

  Comparisons based on Optimization level 40, V4R5. JDK 1.1.8. (* Note that the instance variable was actually a bit faster in the test, but this was judged an artifact of the necessarily simple program used to generate the data -- they should be essentially equal).

  **Note**: This is a performance-oriented suggestion. Making instance variables public reduces the benefit of Object Orientation. Having a local copy in the method of an instance variable can improve performance, but may also add coding complexity (especially in cases where individual blocks use the synchronized keyword). Avoiding the "synchronized" label on a method just for performance may lead to difficult bugs in multithreaded applications.

- *Minimize use of exceptions (try catch blocks)*
  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++) {
        while (true) {
            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 is appropriate in this case.

- *Do not invoke the JAVA/RUNJVA command too often*
  The JAVA/RUNJVA commands create a new batch immediate Job to run the JVM. Limit this operation to relatively long running Java programs. If you need to invoke Java frequently from non-Java programs, consider passing messages through an OS/400 Data Queue. The ToolBox Data Queue classes may be used to implement "hot" JVM's.

- *Explore the General Performance Tips and Techniques in Chapter 20.*
  Some of the discussion in that chapter will apply to Java. Pay particular attention to the discussion "Adjusting Your Performance Tuning for Threads."

- *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 };
```

*Relative Performance*:
When thousands of objects of type test1 and test2 were created, the relative time for test1 was about 5.7 times longer than test2. Since the array myarray2 in class test2 is defined as static final, 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.

Comparisons based on Optimization level 40, v4r5. JDK 1.1.8.

**Java OS/400 Database Access Tips**

- *Use the native JDBC driver*
  There are two OS/400 JDBC drivers that may be used to access local data. Programmers coding connect statements should know that the Toolbox driver is located at Java URL *"jdbc:as400:system-name"* where system-name is the iSeries TCP/IP system name. The native JDBC driver is located at Java URL *"jdbc:db2:system-name"* where the system-name is the Database name. The native OS/400 JDBC driver uses an internal shared memory condition variable to communicate with the SQL/CLI Server Job. The ToolBox JDBC driver assumes that the data is remote and uses a socket connection into the client access ODBC driver. The native JDBC driver is faster when you are accessing local data.

- *Pool Database Connections*
  Connection pooling is a technique for sharing the connection to the OS/400 database between cooperating threads within a JVM. It is useful in many ordinary Java applications, but is especially important in a servlet environment. Since servlets objects are not guaranteed to have a one-to-one correspondence with an invocation of their principal methods (e.g. service() or doGet()), instance variables can't be used as one would ordinarily expect. However, JDBC connections are expensive to create on any platform. A growing literature makes many suggestions about how to "pool" and reuse JDBC connections using either an object associated with each servlet execution instance or via static class functions. WebSphere provides built-in functionality here that is worth mastering. Pooling allows the relatively expensive JDBC connection to be retained for multiple servlet invocations. Perhaps as importantly, it also allows things like PreparedStatement objects to be reused. In a servlet context, this makes the next suggestion much more meaningful.

- *Use Prepared Statements*
  The JDBC prepareStatement method should be used for repeatable executeQuery or executeUpdate methods. If prepareStatement, which generates are reusable PreparedStatement object, is not used, the execute statement will implicitly re-do this work on every execute or executeQuery, even if the effort 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.

  **Note**: Avoid placing the prepareStatement inside of loops (e.g. just before the execute). In some non OS/400 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. It will show a greater improvement in iSeries.

- *Store character data in DB2 as Unicode*
  The OS/400 JVM stores string data internally as 2 byte Unicode. If you are reading or writing large amounts of string data and the data is stored in EBCDIC, the data will be converted on every database access. You can avoid this conversion by storing the data in DB2 as 2 byte Unicode. Use the SQL graphic type with CCSID 13488 or the DDS graphic type with CCSID 13488.

  **Note**: Be careful with this suggestion. 1) If characters are the main portion of the record data, the record could double in size. If this is a large and important database, this will increase hard disk expense, perhaps by a large amount. 2) If the database is accessed by non Java code (e.g. legacy RPG applications) Unicode may create complications for the old code.

- *Store or at least fetch numeric data in DB2 as double*
  Decimal data cannot be represented in Java as a primitive type. Decimal data is converted to the class java.lang.BigDecimal on every database read or write when getBigDecimal is used to access it. BigDecimal is a much more general object and is not really an RPG or COBOL style decimal. The large conversion cost can be avoided by storing or at least fetching numeric data to/from the database as double (e.g. getDouble() or setDouble() on SQL DECIMAL, NUMERIC, FLOAT, and DOUBLE fields). Don't bother with Java float, even for SQL FLOAT as the latter is internally a Java double anyway. Be aware that rounding problems may be introduced with the use of double. In rare cases (e.g. in the banking industry) decimal math can be a requirement. Use BigDecimal for these.

- *Use ToolBox record I/O*
  The OS/400 ToolBox for Java provides native record level access classes. These classes are specific to the OS/400 platform. They provide a significant performance gain over the use of JDBC access. See the AS400File object under Record Level access

- *Consider Using Data Queues to an RPG program or Stored Procedures*
  Especially for very simple database access, dropping out of Java into traditional languages, with native database access, can offer substantial advantages. Having one or more server jobs waiting on a data queue that is accessed by multiple Java threads can be a great way to manage the tradeoff between application performance and multithreaded DB complexity.

- *Check ToolBox for existence of a Java program object*
  The jt400.jar file contains the iSeries ToolBox for Java product. After installation, this .jar file may not have a Java program object. If not, use the CRTJVAPGM at optimization level 40 to create the program object. Use the CRTJVAPGM command during low system activity as it will take some time. Use the DSPJVAPGM command to see if the program object already exists.

**Allocation and Garbage Collection**

- *Set object references to null when done with them.*
  Suppose object A has a reference to object B. Suppose further, that down some code path A no longer needs a reference to B. In that case, one should take the extra trouble to set the variable in A that references B to null. If this is the last reference to B, it can be garbage collected. If it is the last reference and it isn't set null, B will "hang around" instead of being collected. Example: Suppose one codes *myResultSet.close();* In that case, it probably should be followed by *myResultSet = null;*.

- *Leave GCHMAX as default*
  The GCHMAX parameter on the JAVA/RUNJVA command specifies the maximum amount of storage that you want to allocate to the garbage collection heap. In general the default value (Set to the largest possible value) should be used. The system does not allocate additional storage until it is needed. A large value does not impact performance. If a maximum is specified, and reached, the JVM will stop all threads and attempt to synchronously collect objects. If GCHMAX is too small, a java.lang.OutOfMemory error will occur. Some improvements introduced in V4R4 may cause difficulties if GCHMAX was set to a small value in V4R3. Those migrating directly from V4R3 to V4R5 may experience the same problem.

- *Adjust GCHINL as necessary*
  The GCHINL parameter on the JAVA/RUNJVA command specifies the amount of initial storage that you want to allocate to the garbage collection heap. This parameter indirectly affects the frequency of the asynchronous garbage collection processing. When the total allocation for new objects reaches this value, asynchronous collection is started. A larger value for this parameter will cause the garbage collector to run less frequently, but will also allocate a larger heap. The best value for this parameter will depend on the number, size, and lifetime of objects in your application as well as the amount of memory available to your application. Use of OPTION(*VERBOSEGC) can give you details on the frequency of garbage collection, and also object allocation information.

- *Ignore GCHPTY*
  This parameter is not used. It has no effect on performance.

- *Ignore GCHFRQ*
  This parameter is not used. It has no effect on performance.

- *Monitor GC Heap faulting*
  Java objects are maintained in the JVM heap. Excessive page faults may occur if the memory pool for your JVM is too small. These faults will be reported as non-database page faults on the WRKSYSSTS command display. Typically, the storage pool for your JVM is *BASE. Fault rates between 20 and 30 per second are acceptable. Higher rates should be reduced by increasing memory pool size. In some cases, reducing this value below 20 or 30 per second may improve performance as well. If you have the storage available, reducing the rate below 20 to 30 per second may be a benefit.

Lowering the GCHINL parameter might also reduce paging rates by reducing the OS/400 JVM heap size.

- *Minimize Object Creation*
  See previous suggestion about minimizing object creation.

---

## 7.6 Capacity Planning

Java requires more resources than previous languages. Accordingly, when estimating capacity, a more robust machine should ordinarily be specified.

Java's added resources have been diminishing over time (see the previous sections of this chapter).

Still, all in all, it costs more resource to deploy Java than a traditional RPG application today.

We strongly recommend using the Workload Estimator to help you estimate your Java applications. Note that there is a separate section for "ordinary" Java applications and for WebSphere. Each represents differing assumptions. In particular, the WebSphere estimator can take into account the typical added impact of some key WebSphere function such as Java Server Pages (JSPs). A link to the estimator resides at the end of the chapter.

### General Guidelines

None the less, any tool of this kind must necessarily represent averages. Some judgment needs to be applied to the tool's output.

Things to consider when estimating a machine with Java content:

- **First, remember to account for the amount of traditional processing (RPG, COBOL, etc.) going on.** To the extent traditional work is going on, or functions such as SAP are going on, the machine should be sized according to existing capacity planning guidelines. If you are dealing with a new machine, the Workload Estimator has the ability to take estimates for several other kinds of work. The main caveat then becomes Java's growth rate versus the other applications. If Java is the core of someone's e-business and e-business grows rapidly, so will their Java content.

- **Second, be careful to ascertain how much Java is going into the iSeries itself.** If the iSeries is being accessed by client Java code, but the code in OS/400 itself remains in COBOL, RPG, or C/C++, there's no point in padding capacity for Java -- it isn't being used. This is especially important to consider when using Workload Estimator and its WebSphere input display. The Workload Estimator presumes that OS/400 is running WebSphere and sizes accordingly. The important item if the Java function is in the PC becomes ensuring the customer's PCs have enough performance to run Java on the client, and enough traditional horsepower to service their requests. Likewise, if the iSeries is just being used as a Web Server (e.g. Domino GO), there's no need to change capacity planning for Java content for that reason alone. Until Java is used for servlets, an iSeries running WebSphere will not itself be running Java function. The main issue becomes the capacity needed to run general web serving.

- **Third, even when Java servlets and Java applications are being used, account carefully for added system services.** Web serving, communications, and database costs can often swamp Java's

contribution to an end-to-end application. Because it uses JDBC and dynamic SQL, Java can increase the database costs compared to a traditional application doing similar things. When using Workload Estimator, pay particular attention to questions involving database as these questions attempt to balance expected Java servlet/application pathlength against the use of the underlying database.

- **Fourth, recognize that iSeries has been optimized around scalable, OLTP type applications which use lots of system services such as database**. Java, by contrast, will tend to put more of its execution in the application itself. In the short run, simpler servlets may complicate this story, but over time, Java content will grow as a percentage of the processor compared to traditional. The reason relates to Java's portability story. Java will tend to invoke Java-based function where RPG would invoke the operating system. This property will tend to increase processor requirements overall compared to what we're familiar with. Other features of Java will tend to require more main storage than traditional languages. When using Workload Estimator, pay attention to the "complexity" estimate. Here is where the Workload Estimator attempts (in conjunction with the DB questions) to balance application pathlength and database consumption.

- **Fifth, because of the increased processor needs, be wary about using the smallest iSeries models.** This is particularly true of test and development machines. Because of OLTP price/performance tradeoffs, smaller or older machines may be disproportionately disappointing to customers when used for Java, even for testing. In general, make sure the processor performance of the test machine and development machines is in line with that of the production machine. This would mean deploying a machine with a higher uniprocessor CPW rating than would ordinarily be the case. Conversely, if this is not done, do not immediately panic of performance is a bit "off" from what is expected based on results at a development machine. Get some time on the target machine to see if things change for the better.

- **Sixth, beware of misleading benchmarks**. Many individuals will be willing and able to write their own benchmarks for Java. They'll also be able to download some "Java benchmarks" over the Internet. While there is less of this than in former years, this sort of approach is sometimes seen. *Most of these will be poor predictors of server performance*. This includes VolanoMark, which requires careful tuning and primarily measures Communications Performance. Because of this, and Java/400 tradeoffs for better server performance, many of these sorts of benchmarks will also tend to make iSeries look worse than the actual deployment of their application would be. Those running a Java evaluation should make sure that any benchmark: a) is some kind of prototype of a true 'server' application, b) runs long enough (at least 15 minutes) to represent a fair, steady-state comparison, c) has scalability characteristics (multiple threads, multiple Java jobs, etc.). OS/400 Java is not optimized for simple, single-threaded benchmarks. Nor should it be: iSeries customers will tend to deploy multiple servers and threads in a typical Java use (e.g. web serving via servlets). Another thing to watch for: Using an inadequate test machine for benchmarking and then fearing Java isn't acceptable on their bigger, faster production one.

- **Seventh, recognize that Java won't deploy in a traditional manner.** 5250 operations to and from Java will not be a frequent attribute of Java on the OS/400. Accordingly, the higher the Java content, as a percentage of total operations, the more smaller the interactive CPW rating should be.

- **Eighth, consider CIW versus CPW when comparing CPUs** See next section.

- **Ninth, keep in mind that not everything changes for Java.**

1. Whether SMP (Symmetric Multiprocessing) makes sense will not typically change for Java. Java probably will run better with a machine using the fewer CPUs for the same CPW rating, but this is very often true of traditional applications as well.

2. The hard disk (DASD) cost of the database for Java should be about the same. Since database often swamps other uses of DASD, that means that Java should seldom require more disk space than traditional languages. If you are using the Workload Estimator and have good cause to override the estimate, based on a sound understanding of the database cost, you should consider overriding the value. However, for smaller machines, don't forget to make a reasonable allowance for "temporary" storage related to the job and the Java heap before reducing the recommendation.

**CIW versus CPW in Java**

Java is not one monolithic thing. It is a computer language, so applications implemented in Java can vary in their capacities and their use of system resources. However, Java often shares attributes of newer, emerging workloads. That is, Java programs are often computationally intensive. Most traditional commercial applications are a balance between processor and various forms of I/O. Data base access to generate a report is a classic example. However, some modern applications are much more computationally intensive, especially in the application itself.

In V5R1, we have added a new measure called CIW -- Compute Intensive Workload -- to recognize this new reality. This is described in detail elsewhere in this document. Here, the question arises: For a given Java workload -- in and out of WebSphere -- is CPW or CIW the correct way to think about the workload? Note that “workload” here can be an application written in Java or can be a collection of related servlets deployed in WebSphere.

Choose CIW to compare machines when you think or know the workload:

- Spends at least half of the total time in the Java application or prominent Java-based middleware (such as XML).
- Accesses the data base sparingly, if at all.
- Builds large networks of objects within the Java Virtual Machine and consults that network of objects frequently and substantially.

Choose CPW to compare machines when you think or know the workload:

- Seems to resemble, in broad outline, an RPG-like program that sets up a call to the data base, minimally processes each record, and then iterates to do it all over again.
- Uses large, complicated data base SQL queries

For instance, consider a program doing an “insertion sort.” Such a program might build a large network of Java objects (outputting the final network as the sorted output) as it reads each record, one at a time. This program would probably be compared using CPW rather than CIW since its work with the network of objects, will typically be only a handful of objects, since the tree is kept in sorted order as a binary tree.

On the other hand, consider a “search” where, where a large network of preexisting records was read, then new records were compared, one by one, with the preexisting records. If the preexisting records are large enough in number, or the calculations very intense, then CIW would probably be a better choice since many if not most objects in the network would need to be examined.

If, as is usual, you have multiple workloads, then the most prominent workload would be the main one to consider for this question.

**Workload Estimator**

Sizings for selecting the most appropriate iSeries to run Java and/or WebSphere workloads may be estimated using the IBM Workload Estimator for iSeries. The estimator is accessable by anyone at: Http://as400service.ibm.com/estimator .

# Chapter 8. IBM Network Station Performance

Performance information for Releases 1 to 3 for IBM Network Stations attached to V3R2, V3R7, V4R1, V4R2, V4R3 and V4R4 is included below. The following IBM Network Station functions are included:

- Time to initialize the IBM Network Station (prior to login) for ethernet, token ring and twinax
- Time to load the applications (5250 emulation and browser, etc.)
- 5250 application performance
- Browser performance
- Java Virtual Machine (VM) applet/application performance
- Times for the IBM Network Station series 100, 300 and 1000

The computer industry has a generic name for the IBM Network Station - the thin client. Since clients attach to servers, it might seem that an AS/400 SERVER model attached to a Network Station (a thin CLIENT) would always be the best fit (that is, CLIENT to SERVER). Be cautious when the Network Station is attached to an AS/400 SERVER model. When using 5250 applications, the Network Station looks like a Non-Programmable Terminal (NPT) (like an interactive job) to the SERVER and will be subject to the AS/400 SERVER interactive rules, so it might not always be a good fit. The traditional AS/400 SYSTEM models are always a good fit with the Network Station.

In the following sections, references to Release x refer to the Network Station. References to VxRx refer to the AS/400 releases. In addition, in the following sections, performance data for one Network Station is real data; performance data for 10, 50 and 100 Network Stations is simulated. All twinax data is real.

## 8.1 IBM Network Station Network Data

The following table shows the amount of data that flows from the AS/400 to each IBM Network Station for initialization and each application load:

*Table 8.1. Elements Loaded to a Network Station (MB)*

| **Release** | **Rel 1-2.5** | **Rel 1-2.5** | **Rel 3** | **Rel 3 DBCS*** |
|---|---|---|---|---|
| **Series** | 100/300 | 1000 | All | All |
| **Kernel + Configuration + Other** | 4.0 | 4.8 | 3.0 | 3.9 |
| **5250 Emulation** | 0.9 | 0.9 | 1.6 | 3.8 |
| **3270 Emulation** | 0.3 | 0.3 | 0.9 | 3.2 |
| **IBM Network Station Browser** | 2.2 | 2.2 | NA | NA |
| **Navio NC Navigator** | 3.7 | 3.7 | 5.0 | 10.0 |
| **Java Virtual Machine** | 1.5 - 5.0 | 1.5 - 5.0 | 1.5 - 5.0 | 1.5 - 5.0 |

**Note**:
- The amount of data downloaded will vary, depending on the configuration selected
- *DBCS support includes Japanese, Korean, simplified Chinese, and Traditional Chinese

The kernel/configuration data is downloaded when the Network Station is powered-on. Unless configured otherwise, all the other options are downloaded when they are selected.

Note that when an application (e.g. 5250 emulation) is closed or the user logs-out, that application will again be downloaded when it is next selected - it is not kept in memory across log-outs. The kernel/configuration data is kept in the Network Station across log-outs.

The Java Virtual Machine download time varies depending on the application. Only the required classes are downloaded.

In Release 3, some of the information that is sent to the Network Station is compressed. Once received, the Network Station decompresses it. This compression means fewer bits are shipped from the AS/400 to the Network Station, resulting in better LAN utilization. More data/function is shipped to the Network Station in Release 3 than in previous releases. The compression results in boot performance that is about equal to previous releases.

Release 3 contains an option, TFTP subnet broadcast, that can significantly decrease the amount of data transmitted during the boot process, as well as saving significant CPU cycles in the AS/400. This option is described further in the sections below.

---

## 8.2 IBM Network Station Initialization

Initialization, at this time, is not trivial and could be a performance concern for some customers. The time to initialize the Network Station, particularly when many stations are initialized simultaneously can be prohibitive. In addition, initialization can consume a lot of AS/400 CPU, so that other jobs on the AS/400 might be starved.

If possible, it is best to leave the Network Station powered-on after initialization and/or to stagger initialization. The IBM Network Station consumes very little power. If initialization times are a problem and power outages are a concern, battery backups for each IBM Network Station, should be considered or possibly server systems dedicated to initialization.

**Different Initialization Mechanisms - the Gory Details**

Initialization is performed using TCP/IP Trivial File Transfer Protocol (TFTP) and/or AS/400 Remote File System (RFS). Both these access methods read files from the AS/400 to the Network Station. For reliability and performance, both mechanisms subdivide files into blocks for sending, and then recombine them in the Network Station. The TFTP block size can be configured 512 thru 8192 bytes. The RFS block size is fixed at 8192. For Releases 1-2.5, TFTP and/or RFS is used during initialization depending on the configured initialization options. For Release 3, the kernel is loaded using TFTP and then RFS is automatically tried - no matter what the configuration - if RFS is unsuccessful, the configured options are tried.

There are three possible ways to initialize the Network Station:

- NVRAM - the AS/400 and Network Station IP addresses and other information are configured in each Network Station. The Network Station sends a TFTP request to the configured server to begin initialization.

- BOOTP - the Network Station broadcasts to find a responding AS/400 server. The AS/400 server is previously configured with each Network Station's IP address and other information. Once the server receives a broadcast from a Network Station, it sends the configured data to the Network Station and then begins the initialization.

- DHCP - the same as BOOTP except the AS/400 server contains a pool of Network Station IP addresses.

BOOTP or DHCP is the preferred method, for Releases 1-2.5. All methods are OK in Release 3.

For Releases 1-2.5, NVRAM uses TFTP to load the kernel/configuration files and, after login, uses RFS. For Release 3, NVRAM uses TFTP to load the kernel and RFS for all subsequent files.

BOOTP and DHCP use TFTP to load the kernel and then use RFS to load all subsequent files.

For Releases 1-2.5, the Network Station tries 10 times with a 5 second timeout to locate and read the kernel using TFTP. After 10 attempts, an error message is issued. For Release 3, the number of reties can be configured - an infinite retry is preferred.

For Releases 1-2.5, if NVRAM is selected, the Network Station reads the configuration files using TFTP. The Network Station will try 10 times with a 3 second timeout to read each file. If unsuccessful, it will skip that file and then try to read the next file - which eventually results in an unsuccessful initialization. (RFS will not skip files.) From a reliability perspective, this makes NVRAM, for Releases 1-2.5, the least preferred booting mechanism.

Release 3 contains a new option - subnet broadcast. Subnet broadcast is supported on ethernet, token ring and twinax. When this option is selected, TFTP data (the kernel - about 2MB), is broadcast from the AS/400 server to any requesting Network Station. That is, the kernel is sent one time so that each Network Station receives it. When subnet broadcast is off, the kernel is sent individually to each Network Station, which means a lot more data on the LAN/twinax. The broadcast is only to a subnet (e.g. any Network Station on a single ring, such as 9.5.112.x). When Network Stations from different subnets request the kernel, the AS/400 provides a broadcast to each subnet. The data below shows that subnet broadcast uses less AS/400 CPU. Subnet broadcast is the preferred boot option (twinax has some special considerations mentioned below). There is a caution - some routers do not support broadcast and broadcast can cause other problems, if not configured properly.

Subnet broadcast is supported on twinax. Unlike ethernet and token ring, the twinax protocol does not support broadcast. What this means for twinax, is that, when subnet broadcast is selected, each frame is sent individually to each device. When all devices are expecting the broadcast, this option works well (meaning less AS/400 CPU is used). When all devices are not expecting the broadcast, this option is not great (more data on the twinax cable). The data below illustrates this. In general, customers should not use subnet broadcast for twinax.

Some customers who have Series 1000s have experienced performance problems. The Series 1000 supports both full duplex and half duplex. In general, the performance problem is caused by a configuration error. The Series 1000 tries to operate in full duplex mode, but a router or something else in the network supports only half duplex. The Series 1000 almost continuously runs into collisions on the Ethernet which will result in extremely slow performance.

Some customers who have token ring network switches that pass 4K frames have experienced difficulties. The customers had set their LAN frame size/MTU to a value greater than 8K. In general, these customers used NVRAM - with default 1024 TFTP block size. Initialization works fine until login - where RFS takes over and uses 8K frames. The 8K frames do not pass through a 4K switch. Some solutions to this problem might be: configure the switch to allow 8K frames, replace the switch with a router, or configure the AS/400 LAN frame size/MTU to 4K (twinax is fixed at 4K).

If the network has no Domain Name Server (DNS), performance can be very slow. The initialization logic expects a DNS. If none exists, initialization waits for DNS searches to timeout (30 second) before proceeding. AS/400 V4R2 contains DNS support. If a customer does not wish to use a DNS, for Release 3, good performance is still possible by doing the following:

- CFGTCP, Change TCP/IP domain information (option 12), set search priority to *LOCAL
- CFGTCP, Work with TCP/IP host table entries (option 10), add the IP address and host name for the AS/400 and each Network Station
- IBM Network Station Manager, select Hardware, select Workstations, under Domain Name Server, set Update Network Station Manager DNS file

The initialization options described in this white paper will fit most customer environments. There are other variations that can occur. For example, if the customer chooses BOOTP and successfully loads the kernel, but for some reason RFS isn't working properly, initialization will timeout on RFS and switch back to TFTP. Variations such as these are not described in this document. The BOOTP boot sequence is described in greater detail in the following section.

## BOOTP Initialization

There are four steps in the BOOTP initialization process. To get a total initialization time, times from each of the following four steps must be added together:

1. Hardware Test

   The hardware test is just that - a memory test and other hardware tests to ensure that the hardware is operational. For the most part, the length of this test is determined by the amount of memory in the IBM Network Station.

*Table 8.2. Time (Seconds) to Perform Hardware Test*

| Memory (MB) | Series 100 | Series 300 | Series 1000 |
|---|---|---|---|
| 8 | 15 | 14 | -- |
| 16 | 18 | 18 | -- |
| 32 | 24 | 22 | 10 |
| 48 | 30 | 26 | -- |
| 64 | 36 | 31 | 13 |

2. Kernel/Configuration Initialization

   In this step, the Network Station locates the AS/400 server, reads the kernel and configuration files, and then displays the login window.

The Network Station broadcasts a BOOTP request to locate the AS/400 server. Then the kernel (about 2MB) is read using the TFTP function of TCP/IP. And then configuration files are read using the Remote File System (RFS).

The time to load the kernel using TFTP is heavily dependent on:

- TFTP block size
- TCP/IP maximum transmission unit (MTU) size
- LAN line description frame size (fixed for twinax)
- TFTP subnet broadcast option
- number of TFTP jobs

The Network Station negotiates the TFTP block size with the AS/400. It can range from 512 to 8192 bytes. The Network Station default is 8192. In general, the Network Station uses the TFTP block size, MTU and frame size defined by the AS/400.

The AS/400 default TFTP block size is 1024. As will be seen in the following tables, best performance is obtained with a large TFTP block size (e.g. 8192). If the MTU or frame size are less than 8192 (e.g. Ethernet has a maximum frame size of 1492) performance can still be enhanced by configuring the block size greater than the MTU/frame size. If the TFTP block size is greater than the MTU/frame size, TCP/IP fragments (subdivides) the TFTP blocks to fit into the MTU/frame size. The Network Station TCP/IP recombines the MTU/frames into TFTP blocks. This fragmentation provides better performance than setting the TFTP block size equal to the MTU/frame size. Users should be aware that some routers, switches and/or gateways do not support this fragmentation capability. Twinax MTU/frame size are fixed, so fragmentation does not apply to twinax attached Network Stations.

The number of TFTP jobs on the AS/400 is also a performance factor - the optimal number for a system with a single LAN IOP is about 6, the default. The TFTP jobs are a pool of AS/400 jobs that download the kernel to Network Stations. They are first come, first serve. If there are more Network Station requests than jobs, the excess are ignored (i.e. not queued). If a request is not satisfied, the Network Station, every 5 seconds, will repeat its request. In general, there should be 6 TFTP jobs for each LAN IOP that has attached Network Stations.

The following tables and figures show how the TFTP block size affects the kernel/configuration initialization time, for a few AS/400 system sizes. The tables also show what happens when 1, 10, 50, and 100 Network Stations simultaneously (e.g. after a power outage) request TFTP initialization. The times represent the number of seconds when the last Network Station completes its TFTP and RFS download.

The data in the following tables was obtained in a dedicated environment. That is, only BOOTP, TFTP and RFS were running on the AS/400 and there was no other load on the LAN. In each test case, the base pool (memory) was cleared before beginning the test.

Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here.

*Table 8.3. Kernel/Configuration - AS/400 F97 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model F97 (V3R2)**
**IBM Network Station Series 300 (Releases 1-2.5)**
**16Mb Token-Ring**
**8KB MTU/Frame Sizes, 6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 109 (5.0) | 46 (5.5) | 34 (4.2) | 29 (2.9) | 26 (2.6) |
| **10** | 225 (27.0) | 105 (31.0) | 77 (22.6) | 63 (17.1) | 57 (12.2) |
| **50** | 992 (32.8) | 470 (41.7) | 327 (30.8) | 257 (24.0) | 209 (20.1) |
| **100** | 1885 (35.2) | 890 (46.3) | 624 (33.6) | 503 (25.5) | 395 (22.3) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.4. Kernel/Configuration - AS/400 150-2270 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model 150-2270 (V3R7)**
**IBM Network Station Series 300 (Releases 1-2.5)**
**16Mb Token-Ring MFIOP**
**8KB MTU/Frame Sizes, 6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 85 (23.3) | 35 (28.6) | 31 (22.0) | 27 (16.3) | 26 (14.8) |
| **10** | 229 (87.8) | 126 (82.2) | 83 (72.9) | 63 (63.4) | 55 (53.6) |
| **50** | 1065 (94.2) | 565 (95.0) | 347 (92.0) | 234 (87.6) | 193 (77.6) |
| **100** | 2075 (97.5) | 1119 (97.0) | 682 (94.5) | 448 (92.5) | 352 (88.1) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.5. Kernel/Configuration - AS/400 510-2144 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model 510-2144 (V3R7)**
**IBM Network Station Series 300 (Releases 1-2.5)**
**2619 16Mb Token-Ring LAN IOP**
**8KB MTU/Frame Sizes, 6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 71 (9.8) | 59 (7.4) | 52 (6.4) | 46 (5.8) | 43 (5.2) |
| **10** | 169 (39.3) | 117 (30.3) | 81 (26.1) | 65 (21.2) | 62 (17.3) |
| **50** | 790 (44.5) | 451 (42.4) | 361 (32.6) | 265 (28.7) | 209 (27.0) |
| **100** | 1526 (47.3) | 875 (45.2) | 667 (35.7) | 498 (31.7) | 384 (30.5) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.6. Kernel/Configuration - AS/400 S30-2257 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model S30-2257 (V4R1)**
**IBM Network Station Series 300 (Releases 1-2.5)**
**2629 16Mb Token-Ring LAN IOP**
**8KB MTU/Frame Sizes, 6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 96 (1.8) | 41 (4.3) | 33 (4.5) | 30 (3.7) | 29 (3.6) |
| **10** | 182 (14.4) | 73 (16.4) | 56 (12.7) | 52 (8.9) | 39 (8.5) |
| **50** | 735 (18.9) | 279 (24.9) | 201 (20.2) | 146 (17.5) | 127 (15.3) |
| **100** | 1382 (20.2) | 513 (27.7) | 357 (23.2) | 272 (20.0) | 244 (16.6) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.7. Kernel/Configuration - AS/400 400-2132 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model 400-2132 (V4R1)**
**IBM Network Station Series 300 (Releases 1-2.5)**
**2629 10Mb Ethernet LAN IOP**
**6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 76 (35.6) | 53 (26.3) | 45 (19.8) | 39 (17.7) | 34 (15.5) |
| **10** | 280 (90.2) | 167 (82.0) | 110 (72.6) | 83 (63.7) | 67 (55.7) |
| **50** | 1311 (97.5) | 745 (93.8) | 467 (88.6) | 321 (82.1) | 277 (69.3) |
| **100** | 2591 (97.8) | 1466 (96.9) | 895 (93.4) | 623 (86.7) | 540 (73.1) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.8. Kernel/Configuration - AS/400 400-2132 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model 400-2132 (V4R2)**
**IBM Network Station Series 300 (Release 3)**
**2629 10Mb Ethernet LAN IOP**
**6 TFTP Jobs**
**Vary TFTP Block Size**

| # NS | 512 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|
| **1** | 64 (33.1) | 51 (22.9) | 46 (15.9) | 42 (13.4) | 40 (10.6) |
| **10** | 345 (82.8) | 200 (75.0) | 122 (62.4) | 89 (51.1) | 72 (48.4) |
| **50** | 1831 (93.3) | 1109 (89.9) | 533 (85.6) | 334 (81.7) | 274 (67.6) |
| **100** | 3678 (93.9) | 1985 (90.1) | 1111 (88.6) | 660 (85.2) | 543 (72.1) |

**Note**:
- Results may differ significantly from those listed here

*Table 8.9. Kernel/Configuration - AS/400 400-2132 and Network Station 300*

**Kernel/Configuration Initialization Time in Seconds**
**(Average CPU Utilization in %)**
**AS/400 Model 400-2132 (V4R2)**
**IBM Network Station Series 300 (Release 3)**
**All NSs Attached to a Single Twinax Adapter**
**6 TFTP Jobs**
**Vary Twinax Adapter Type, Subnet Broadcast Option and TFTP Block Size**

| # NS | 6050 w/o Subnet 8K TFTP | 6180 with Subnet 1K TFTP | 6180 w/o Subnet 1K TFTP | 6180 with Subnet 8K TFTP | 6180 w/o Subnet 8K TFTP |
|---|---|---|---|---|---|
| **1** | 107 (8.5) | 114 (22.0) | 116 (22.1) | 87 (14.5) | 82 (13.3) |
| **2** | 173 | 155 | 133 | 90 | 85 |
| **3** | 225 | 165 | 154 | 106 | 98 |
| **4** | 275 | 168 | 159 | 121 | 116 |
| **5** | 325 | 186 | 178 | 142 | 139 |
| **6** | 388 | 201 | 199 | 155 | 157 |
| **7** | 446 (16.4) | 225 (33.6) | 221 (70.0) | 171 (28.8) | 162 (37.5) |

**Note**:
- Results may differ significantly from those listed here

Note that subnet broadcast uses less AS/400 CPU. However, as discussed above, each twinax device on the subnet will get their own copy of the broadcast data, even if they didn't request it, which would mean unwanted data on the twinax cable. In general, customers should not use twinax subnet broadcast. (Subnet broadcast should be used on LANs.)

In Table 8.9, the Network Stations were all chained to a single cable port. For the 6180 adapter, faster times could be obtained if the Network Stations were balanced across cable ports, that is, half on ports 0-3 and the other half on ports 4-7. For example, in the table above, 6 Network Stations with an 8K TFTP block size, without subnet broadcast, booted in 157 seconds. If they had been balanced, 3 on port 0 and 2 on port 4, the initialization time would have been 130 seconds, 17% faster.

If a Network Station has multiple paths, with the same network address, to an AS/400 (e.g. two IOPs that each have a path to the Network Station), unexpected results may occur. Whenever the AS/400 gets a request from a Network Station, it uses the default path to get back to the requesting station. The return route (and any subsequent request/replies) may be different from the original request. This implies that there is no value to add a second IOP with the same network address to gain additional TFTP performance.

TFTP jobs are assigned first come, first serve. There is no mechanism to allocate a TFTP job to a particular IOP. This implies that it is possible for Network Stations attached to one network to monopolize all the TFTP jobs until completion of the kernel download. Other IBM Network Stations maybe starved until a TFTP job is available.

3. Login

Login is just that - the user enters his/her user-ID and password and then the desktop appears.

The load times can be found in the table below.

4. Application Load

Applications are loaded when their respective desktop buttons are selected. Load times vary by AS/400 machines size.

Getting to a 5250 sign-on can require two steps - from the menu bar, select the 5250 button to get to the host name window and then enter the desired host name to get to the 5250 sign-on window. Most administrators use the Network Station Manager to configure for direct menu bar to 5250 sign-on.

Getting to the browser is a single step - from the menu bar, select the browser button to get to the browser.

Examples of load times can be found in the table below.

*Table 8.10. Application Load Times*

**Load Times in Seconds**
**AS/400 Model 150-2270 and 510-2144 (V3R7)**
**IBM Network Station Series 100 and 300 (Releases 1-2.5)**
**2619 16Mb Token-Ring LAN IOP**

| | **2270 100** | **2270 300** | **2144 100** | **2144 300** |
|---|---|---|---|---|
| **Login to desktop** | 10 | 10 | 15 | 11 |
| **5250 select to host name** | 9 | 6 | 10 | 7 |
| **Host name to 5250 login** | 6 | 6 | 12 | 11 |
| **Browser select to browser** | 33 | 16 | 41 | 22 |

**Note**:
- Results may differ significantly from those listed here

*Table 8.11. Application Load Times*

**Load Times in Seconds**
**AS/400 Model 400-2132 (V4R2)**
**IBM Network Station Series 300 (Release 3)**
**eSuite is IBM Network Station 1000 (Release 3)**
**Twinax or Ethernet Adapter**

| | **6050** | **6180** | **2629** | **2629 DBCS*** |
|---|---|---|---|---|
| **Login to desktop** | 30 | 27 | 18 | 23 |
| **5250 select to host name** | 57 | 33 | 10 | 19 |
| **Host name to 5250 login** | 15 | 21 | 12 | 14 |
| **Browser select to browser** | 169 | 131 | 41 | 52 |
| **eSuite select to eSuite** | -- | -- | 175 | -- |

**Note**:
- Results may differ significantly from those listed here
- *DBCS support includes Japanese, Korean, simplified Chinese, and Traditional Chinese

Another example of subnet broadcast: Assume 100 Series 300 Network Stations attached to an AS/400 V4R2 2132 via a single 10Mb ethernet segment. Assume the electricity on all 100 Networks Stations goes out and a while later comes back on. Assume the Network Stations all

have the same memory size (e.g. 32MB) and identical monitors attached. It would be possible for all 100 to be at the Login window in 280 seconds (less than 5 minutes). The 280 seconds comes from: 21 seconds for hardware test, 30 seconds to load the kernel, and 229 seconds to load configuration files.

## 8.3 AS/400 5250 Applications

The Network Station user should see 5250 applications almost exactly as with NPT or PC terminals. However, the load on the AS/400 may be different. Network Stations use the AS/400 TCP/IP Telnet path. Telnet consumes 27% more CPU time per transaction than an NPT attached to a local twinax for a typical commercial workload. This yields a 20% capacity reduction over a twinax attached NPT. For comparison, a Client/Access PC using 5250 over SNA, when using the same workload, consumes 10% more CPU time per transaction than a local twinax attached NPT.

The implication is that customers migrating from local twinax attached NPTs to LAN attached Network Stations will probably use more CPU to run the same 5250 applications. Customers migrating from LAN attached SNA Client/Access PCs will also probably use more CPU. Customers migrating from LAN attached TCP Client/Access PCs should need no additional CPU capacity to run their 5250 applications.

## 8.4 Browser

In general, the Series 100, 300 and 1000 all perform equally well. Their performance should be comparable to that seen on a PC.

It is important that either socks or proxy are configured, but not both. Poor performance is seen when both are used.

Disk caching should never be used.

## 8.5 Java Virtual Machine Applets/Applications

Java is still evolving. As such, its use on a Network Station is also evolving. The Series 100 clearly should not be used for Java. The Series 300, while twice as fast as the 100, can be used for very limited Java applets. The Series 1000 is for Java; however, since Java has varied uses, customers are encouraged to test their Java applications on the Series 1000 before putting them into production.

## 8.6 The AS/400 as a Router

The AS/400 is a router (data passes though it) when twinax attached Network Stations send/receive data from the internet or other servers. At this time, limited performance data is available. The following two tables show results when data is read from an NT server through an AS/400 to a Network Station.

*Table 8.12. LAN to LAN Throughput*

**LAN to LAN Throughput**
**AS/400 Model 400-2132 (V4R2)**
**IBM Network Station Series 300 (Release 3) via 10Mb Eth to AS/400**
**300MHz PC NT server via 16Mb TR to AS/400**
**2629 LAN IOPs, 15MB of Data, 8K TFTP Block**

| # NS | Time (sec) | AS/400 Util (%) | AS/400 Throughput (Kb/s) |
|---|---|---|---|
| **1** | 44 | 11.2 | 340.9 |
| **2** | 48 | 16.9 | 625.0 |
| **3** | 57 | 18.1 | 789.5 |
| **4** | 71 | 25.7 | 845.1 |
| **5** | 90 | 24.4 | 833.3 |
| **10** | 158 | 29.9 | 949.4 |
| **15** | 232 | 34.9 | 969.8 |

*Table 8.13. LAN to Twinax Throughput*

**LAN to Twinax Throughput**
**AS/400 Model 400-2132 (V4R2)**
**IBM Network Station Series 300 (Release 3) via Twinax to AS/400**
**300MHz PC NT server via 16Mb TR to AS/400**
**2629 LAN IOPs, 6180 Twinax Adapter, 2MB of Data, 8K TFTP Block**

| # NS | Time (sec) | AS/400 Util (%) | AS/400 Throughput (Kb/s) |
|---|---|---|---|
| **1** | 33 | 9.9 | 70.1 |
| **2** | 48 | 9.9 | 96.4 |
| **3** | 109 | 10.3 | 63.7 |
| **4** | 127 | 10.5 | 72.9 |
| **5** | 150 | 11.1 | 77.1 |
| **6** | 213 | 11.0 | 65.2 |

## 8.7 Conclusions

The IBM Network Station provides for an excellent working environment.

- In general, the Network Station 1000 performs better than the 300 which performs better than the 100
- Initialization
  - ❖ The Network Station Series 1000 initialization time is about the same as the 300, except for hardware test, where the 1000 is faster. The 300 is faster than the 100.
  - ❖ If possible, customers should consider a boot server for each ring or ethernet.
  - ❖ For Releases 1-2.5, customers should use BOOTP or DHCP and not NVRAM. BOOTP and DHCP are faster and more reliable. For Release 3, all three initialization mechanisms are equal in reliability and performance. BOOTP is slightly (1-2 seconds) faster than DHCP.
  - ❖ The time to initialize Network Stations depends on many variables, such as size of AS/400, TFTP block size, number of attached IBM Network Stations, LAN utilization, CPU utilization, etc. Customers will need to evaluate their own needs. It is recommended that customers go slow in

building their Network Station solutions.

- Initialization time varies from AS/400 model to AS/400 model. In general, the faster the model, the better the performance. On faster models, the bottleneck is the LAN IOP and, on slower models, the bottleneck is CPU and LAN IOP. The 2629 LAN IOP provides better performance than the 2619.

- 10Mb Ethernet, 100Mb Ethernet and 16Mb token ring are about equal.

- During initialization, CPU utilization can be quite high, especially on the smaller AS/400s, which will impact other jobs. In addition, TFTP requires more CPU than RFS.

- Subnet broadcast can significantly reduce LAN traffic and AS/400 CPU utilization. Subnet broadcast is available with AS/400 V4R2 and Network Station Release 3. If possible, it is highly recommended that subnet broadcast be used. In general, subnet broadcast is not advisable with twinax, except as discussed earlier.

- The network administer should configure TCP/IP, LAN frame size and TFTP block size for best performance. In general, the larger the size, the better the performance.

- For twinax, the 6180 adapter is significantly faster than the 6050. The 6180 is about equal to a 4Mb token ring.

- There is no value to add a second IOP, with the same network address, to a LAN to get better initialization performance, since TFTP will select the path to be used. All Network Stations, from the same network, will use the TFTP selected path.

- It is best to configure 6 TFTP jobs per LAN that has attached Network Stations. However, for systems that have multiple LANs, since there is no way, at this time, to dedicate a TFTP job to a particular LAN, initialization may not perform as well as desired.

- In general V4R2 provides better performance than V4R1, which provides better performance than V3R7. V4R2 contains TCP/IP and IOP LAN enhancements. In some cases, customers will see substantial improvements in kernel/configuration initialization. These improvements, in general, will be visible when a single Network Station is initialized with a small TFTP block size. V4R2 contains RFS enhancements. V4R3 and V4R4 performance is the same as V4R2.

- Release 3 boots about as fast as previous releases, even though more data and function are sent. Much of the data sent is compressed.

- Switches, routers and gateways can cause problems. It is best to have a network administrator.

- For 6180 twinax attached Network Stations, best performance is obtained if all Express Datastream enabled devices are on the same cable, excluding older, non-Express capable devices.

- When Express devices are attached to a single workstation controller, best performance is obtained by load balancing those devices. That is, half the devices should be connected to cable ports 0-3, and the other half should be connected to ports 4-7.

- 5250 application performance on the AS/400
  - ❖ In general, the 100, 300 and 1000 all perform equally.
  - ❖ Customers migrating from LAN attached SNA Client/Access PCs will probably use more CPU (about 17%) to run the same 5250 applications.
  - ❖ Customers migrating from LAN attached TCP/IP Client/Access PCs will use about the same CPU to run the same 5250 applications.
  - ❖ Customers migrating from local twinax attached NPTs to IBM Network Stations will probably use more CPU (about 27%) to run the same 5250 applications.
- Browsers
  - ❖ In general, the 100, 300 and 1000 all perform equally.
  - ❖ Poor performance is obtained when both socks and proxy are configured. Only one should be used.
  - ❖ Never use disk caching.
- Java Virtual Machine
  - ❖ The Series 100 should not be used for Java.
  - ❖ The Series 300 can be used for limited, lightweight Java
  - ❖ The Series 1000 is for Java; however, since Java hasn't fully matured and can be used for many, varied applications, customers should insure that their Java application and the 1000 are compatible.
- AS/400 as a Router
  - ❖ Limited performance data is available. A model 400-2132 is able to route about 970kb/s from one LAN to another and about 75Kb/s from a LAN to twinax.

# Chapter 9. AS/400 File Serving Performance

This chapter will focus on AS/400 File Serving Performance.

## 9.1 AS/400 File Serving Performance

When going from **V4R5 to V5R1**, customers will see little or no improvement for AS/400 File Serving Performance.

In V4R4, performance improvements were made to the Integrated File System (IFS). The V4R4 enhancements affect the Root, QOpensys, and User-Defined File Systems. The other file systems (Qsys,QDLS, Qopt...) will function at the same level of performance as the previous release.

The Pre-bring buffering schemes were improved, along with other changes, resulting in WRITES being upto 2x faster. As the number of files being accessed and the size of these files increase, the degree of improvement decreases to the point where there might not be any noticeable change in performance.

## 9.2 AS/400 NetServer File Serving Performance

AS/400 NetServer supports the Server Message Block (SMB) protocol through the use of Transmission Control Protocol/Internet Protocol (TCP/IP) on AS/400. This communication allows clients to access AS/400 shared directory paths and shared output queues. PC clients on the network utilize the file and print-sharing functions that are included in their operating systems. You can configure AS/400 NetServer properties and the properties of AS/400 NetServer file shares and print shares with Operations Navigator.

Clients can use AS/400 NetServer support to install Client Access from the AS/400 since the clients use function that is included in their operating system. See http://www.as400.ibm.com/netserver/ for additional information on AS/400 NetServer.

In V4R4**,** enhancements were made to AS/400 NetServer . The V4R4 enhancements affect the dally timer delay cycle, and TCP Send and Receive buffer sizes. When a user migrates from Network Drives with Client Access to AS/400 NetServer on V4R4, they can expect to see an improvement in performance**.**

**V4R5 Note:** Tables and Figures were added to reflect the V4R5 numbers in comparison to V4R4. The Server and Client specifications stated below were used for both V4R4 and V5R4 performance measurements.

### *AS/400 NetServer Performance*

***Server***

- AS/400 Invader Model 2292/170 V4R5
  1 GB Memory, 935,330 MB for Base Pool, 9- 8GB Disk Drives
  16Mbps Token Ring LAN - 2724 IOP

***Client***s

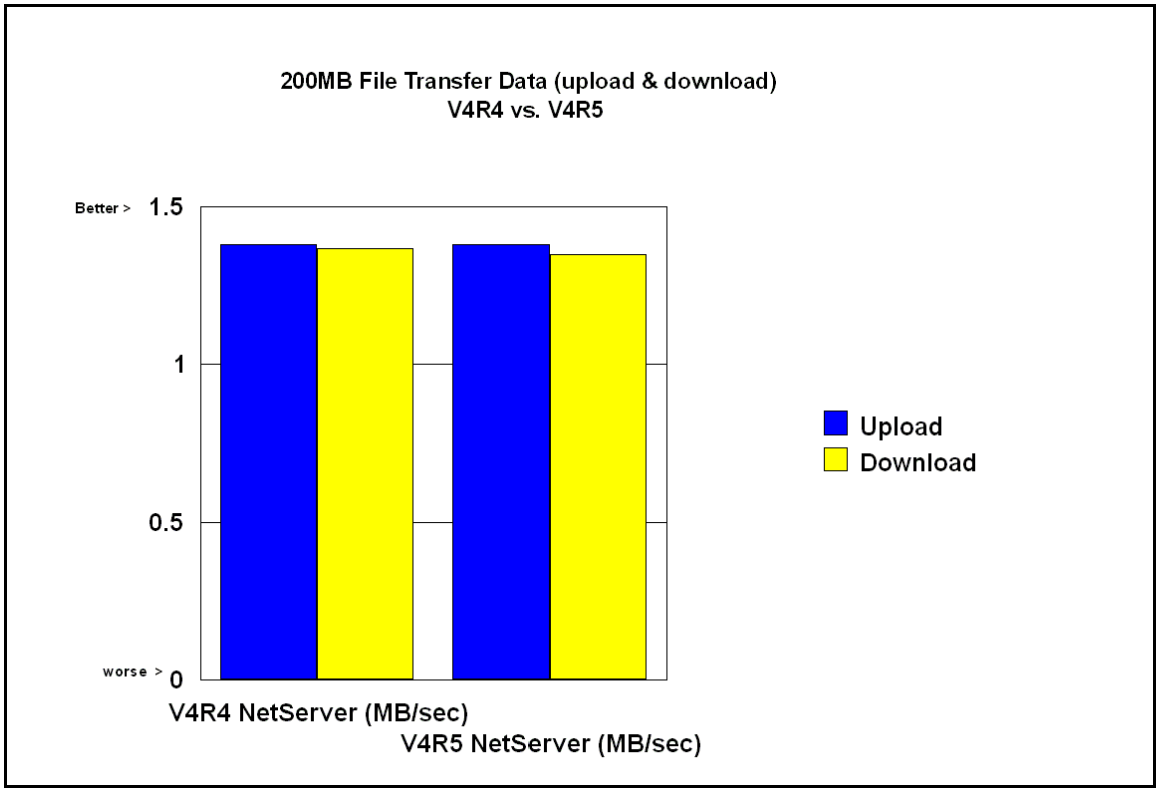
- Pentium 133MHz, 32 MB Memory, 1.2GB IDE Disk Drive, 16Mbps Token Ring PCI Adapter
  Client Access Express for Windows NT (This product was installed for comparison purposes only to provide a comparable PC environment. Connection of the network drive to AS/400 NetServer was done using the function provided with Windows NT.)
  Windows NT Workstation 4.0

- Pentium 133MHz, 32 MB Memory, 1.2GB IDE Disk Drive, 16Mbps Token Ring PCI Adapter
  Client Access for Windows NT V3R1M2
  Windows NT Workstation 4.0

***Workload***

200MB File Transfer in both directions ( Upload and Download using DOS copycommand)

**Measurement Results**:



**Conclusion/Explanations**:

From the chart above in the Measurement Results section, it is evident that when customers migrate from V4R4 to V4R5 they can see a little or no change in performance. It is also clear from the chart that AS/400 NetServer achieves about the same performance when uploading large file to AS400 where as when downloading large files to the AS400 the performance is sightly decrease in performance.

# Chapter 10. DB2/400 Client/Server and Remote Access Performance

With V5R1, overall system performance of the high-end iSeries models increased significantly, giving outstanding growth and improved price/performance. For customers who have a mixed environment (a combination of fixed function workstations and PC's), then the new iSeries models provide significant system performance growth and improved price/performance (see Chapter 2, “AS/400 System Capacities and CPW” for relative performance of these models).

When using client/server technology, it is important to consider the impact of the various client and server components, and their effect on performance. There are different ways of implementing client/server applications. In this chapter, guidelines are provided for a number of common implementation strategies, to help understand the impact of the performance of the client system and the server system using database serving workloads.

With the introduction of Java on the AS/400, database access and client/server development have changed to an even more open environment. The AS/400 Toolbox for Java provides a JDBC driver that conforms to the JDBC specification published by Sun Microsystems. JDBC enables application developers to write portable applets and applications that access relational database information.

Client Access/400 contains an OLE/DB driver for the AS/400 in V4R1. This driver allows developers to easily and quickly develop client/server applications for the AS/400.

ODBC and JDBC will continue to be strongly supported by IBM as an ***open*** way of connecting to DB2/400.

In **V5R1**, NetServer provides file serving performance comparable with that of V4R5.

In V5R1, NetServer provides file serving performance comparable with Network Drive when comparing different types of AS/400 with the same CPW value.

Use the information provided in *AS/400 Performance Capabilities Reference (V3R2), ZC41-8166*, Chapter 7, "DB2/400 Client/Server and Remote Access Performance Information", as a guide for V3R2 performance. In addition, refer to “Related Publications/Documents” at the beginning of this document on how to access a presentation that covers *AS/400 Versus Microsoft's SNA Server Gateway*.

## 10.1 Client Performance Comparisons

Under different client server implementations, PC hardware configuration plays an important role in overall performance. In general, a faster PC CPU will improve performance, but there are other issues which should be taken into account such as disk drive performance, main storage, memory cache etc. If an acceptable relative performance value was 2.0 (or 2X slower than a '486 @ 66 MHZ), then the PS/2 model 80 would prevent this environment from achieving the required response time criteria.

For applications where 50% or more of the application response time contribution is on the PC client, such as a query download or an OLTP application, better performance can be achieved by focusing on the client performance and selecting a faster 486, Pentium (**), Pentium Pro (**), or Pentium II (**) processor.

It is also important to note that client memory can be a significant component of response time as well. For the Windows 3.1 client, most database serving operations will perform acceptably if the client has at least 8 MB of memory. The Windows 95 client performs acceptably with 32MB of memory, and the Windows NT client performs acceptably with 32MB of memory. Client/Server operations that operate on a large amount of data will usually perform better if the client has more than the amount of memory suggested above.

When most of the response time contribution is on the AS/400 such as a complex query, or when processing an OLTP stored procedure, greater performance improvements may be achieved by optimizing the AS/400 application or upgrading AS/400 hardware. For example, it may be possible to create a logical view for a query which is frequently executed.

## 10.2 AS/400 Toolbox for Java

The AS/400 Toolbox for Java is a set of enablers that supports an internet programming model. It provides familiar client/server programming interfaces for use by Java applets and applications. The toolbox does not require additional client support over and above what is provided by the Java Virtual Machine and JDK.

The toolbox provides support similar to functions available when using the Client Access/400 APIs. It uses sockets connections to the existing OS/400 servers as the access mechanism for the AS/400 system. Each server runs in a separate job on the AS/400 system and sends and receives architected data streams on a socket connection.

The AS/400 Toolbox for Java is delivered as a Java package that works with existing servers to provide an internet-enabled interface to access and update AS/400 data and resources.

The base API package contains a set of Java classes that represent AS/400 data and resources. The classes do not have an end-user interface but simply move data back and forth between the client program and an AS/400 system, under the control of the client Java program.

For more information on the AS/400 Toolbox for Java see the Redbook *Accessing the AS/400 System with Java* Document Number SG24-2152-00.

### JDBC Driver

The JDBC driver that is included in the AS/400 Toolbox for Java allows database access to the AS/400 using APIs that are similar to ODBC. This JDBC driver talks to the same server job on the AS/400 as the ODBC driver included with Client Access/400. Many of the options for the Client Access/400 ODBC driver are therefore included in the JDBC driver. Also, any of the database and communication tuning for ODBC and Client Access/400 can be used for JDBC and the Toolbox.

JDBC allows SQL statements to be sent to the AS/400 system for execution. If an SQL statement is run more than one time, use a PreparedStatement object to execute the statement. A PreparedStatement compiles the SQL once, so that subsequent executions run quickly. If a plain Statement object is used, the SQL must be compiled and run every time it is executed. Use Extended Dynamic support; it caches the SQL statements in SQL packages on the AS/400 system. Also turn on package cache; it caches SQL statements in memory.

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. In a client/server environment, stored procedures can help reduce communication I/Os and thus help improve response time.

Use a just-in-time (JIT) compiler for your Java execution environment if possible. The latest JIT technology allows Java programs to perform almost as well as native code written in C or C++. Users of the AS/400 Toolbox for Java can expect the JDBC driver to perform almost as well as a C++ application using ODBC for OLTP types of applications. JDBC applications that download larger amounts of data will perform slower than a comparable C++ application in ODBC because of the object orientated design of the JDBC driver.

There are many properties that can be specified on the JDBC URL or in the JDBC properties object. Several of these properties can significantly affect the performance of a JDBC client/server application and should be utilized where possible. The properties control record blocking, package caching, and extended dynamic support. See the JDBC driver documentation for details on setting these properties. Most of the properties have close parallels in the ODBC driver. Tuning advice from ODBC can be used for JDBC when setting these values.

**Record Level Access**

AS/400 physical files can be accessed a record at a time using the public interface of these classes. Files and members can be created, read, deleted, and updated. The record format can be defined by the programmer at application development time, or can be retrieved at runtime by the AS/400 Toolbox for Java support. These classes use the DDM server to access the AS/400 system. To use the host DDM server through a TCP/IP interface, some special PTFs are required. Check the AS/400 Toolbox for Java documentation for the latest PTF numbers and set up instructions.

Record Level Access can offer better performance than JDBC for applications that need to process AS/400 database data one record at a time. Record access does not go through the SQL query processing that JDBC must go through to process data, therefore it can retrieve a single record quicker than JDBC. However, if complex computations or large sets of records are processed, JDBC may be a better solution. Use JDBC when an SQL statement can be built that does most of the work on the server, or when you want to limit which fields in a record get transfered to the client. The current JDBC specification does not have a mechanisim to insert multiple records at a time (e.g., ODBC's Blocked Insert). Therefore, use the support in Record Level Access to write multiple records at once.

There are several issues that should be considered when using the Record Level Access support in the Toolbox. First, when accessing a file, if the program is going to use the file multiple times, the file should be left open inbetween operations to avoid the extra processing due to open and close. Second, the block size that is specified on the open method should be selected acording to the type of access in the file. Block size is the number of records to download to the client when reading. If multiple records that are relatively close together are going to be retrieved then a block size that can transmit all of the records at once is prefered. However, do not select a block size that will cause a large delay when downloading (e.g., a 1MB download). If the type of access for the file is random, and the records retrieved are not close together in the file, a block size of 1 is prefered. Third, when downloading an entire file that is relatively small, use the

readAll method. This method is considerably faster when reading small files, large files (e.g. >1MB) may encounter "Out of memory" errors, beacuse the entire file is placed and translated in the client's memory.

## 10.3 Client Access/400

### Client Access/400 Express for Windows (XE1)

Client Access/400 Express is the premier client connectivity product for iSeries. Client Access/400 Express communications support provides excellent performance with native TCP/IP connectivity. .

### OLE/DB and ADO Data Access (Project Lightning)

The OLE/DB driver that has been added to the base Client Access/400 for Windows 95/NT allows database access to the AS/400. Developers can write applications that use the OLE/DB driver to access the AS/400 database through DDM Record Level Access, SQL, Stored Procedures, etc. These interfaces can be easily programmed through the ADO layer that most current development environments support (e.g., Visual Basic, Delphi, etc).

The current ADO specification does not support record blocking; therefore, downloading a large table through record level access may take longer than other methods (e.g., ODBC which has record blocking) and consume more network resources, since each record is transmitted as one communication. Look for future versions of the ADO specification to contain record blocking.

The SQL support in the OLE/DB driver uses the same server program as the ODBC driver. This means that developers can use some of the same techniques and ideas from the Client Access ODBC driver for the OLE/DB driver. One important performance improvement that developers can use, is implement prepared statments if an SQL statement is to be executed more than once. Also when executing a prepared statment set the third parameter to be -1, otherwise ADO assumes this is a new statement and discards the previously prepared statement.

### Open Data Base Connectivity - ODBC(**)

In V3R1, the ODBC APIs were significantly enhanced in terms of function and performance compared to the original version in V2R3 (see *AS/400 Performance Capabilities Reference (V3R2), ZC41-8166*, Chapter 7, for more details). ODBC support for the Windows (**) 3.1/95/NT clients in Client Access/400 provides superior performance over the original Remote SQL support in Client Access/400 and its predecessor PC Support/400.

ODBC is a set of API's (Application Programming Interfaces) which provide clients with an open interface to any ODBC supported database. AS/400 supports ODBC with the Windows 3.1, Windows 95, Windows NT and OS/2 client support in Client Access/400. Customers can purchase ODBC drivers to connect to their system(s) and either write applications which utilize the ODBC APIs or purchase an existing application which utilizes the ODBC APIs.

### Client/Server 4GL and Middleware

Many users build client/server applications using client toolkits such as C/S fourth generation languages (4GLs) . Most of the new 4GL tools use "middleware" or interface code to connect to a server. This middleware usually consists of one or more DLLs used to connect to a given server. The middleware converts the client's request into commands and data which the server can understand and converts the server's response into commands and data which the client can understand. Often the middleware is written by the toolkit provider to interface to a given server or to a standard server API set.

Examples of 4GL toolkits are GUPTA's SQLWindows**, PowerSoft's Powerbuilder**, Microsoft's VisualBasic**, and Visual C ++. Examples of middleware standards are Microsoft's ODBC standard and IBM's DRDA standard.

Because the user is often isolated from the APIs and the middleware manages the database access method, it is important to build applications using tools that optimize for performance. In many cases, tools that are built for "openness" for many servers tend to be the worst performers because they are built to the least common denominators. The AS/400 supports many features that enhance performance. Ensure that your toolkit has support for functions like stored procedures and blocked insert. If not, ensure that there is a mechanism to write directly to the CA/400 API set for the best performance. For more information on client/server application development tools, see *AS/400 Client/Server Performance Using Application Development Tools, (SG24-4731).*

### Client Toolkit ODBC Performance

Many performance problems with client development toolkits are due to the client tool creating inefficient database access requests. For example, a simple database transaction that should result in minimal interaction with the server can generate hundreds of unnecessary ODBC requests and responses. By choosing high performance toolkits and with planning and tuning, these problems can be avoided.

Tools are available to diagnose and debug problems with client toolkits and applications. Use tools such as ODBCSpy or ODBC Trace (available through the ODBC Driver Manager) to verify the efficiency of the SQL and ODBC calls that are generated. Also, the toolkits themselves often have tools to trace their server access methods.

## Client/Server Online Transaction Processing (OLTP)

OLTP applications are typically designed for business computing. An OLTP transaction usually consists of several database operations and related computations. Performance requirements for this type of transaction are usually stringent. In the client/server environment an OLTP transaction typically consists of several requests/responses between the client and the server, resulting in a small to moderate amount of data transferred to the client. Because these transactions tend to be repetitive, they are good candidates for application serving (such as remote procedures or distributed processing). It is especially important to avoid unnecessary overhead in processing these repetitive transactions (such as PREPARE operations). Use of parameter markers, stored procedures, triggers, and Extended Dynamic (package) support is recommended to improve the performance of this class of queries.

**Server Challenge Benchmark (SCB)**

***SCB Overview***: The Server Challenge Benchmark (SCB) is a set of three individual workloads developed by IBM to compare the AS/400 against the competition:

1. Transaction-based Component Workload

2. Decision Support Component Workload

3. File Server Component Workload

Each of these component workloads were built to use client/server characteristics. For a detailed description of SCB refer to the IBM white paper titled "AS/400 Client Server Performance Benchmark Guide" available on HONE.

This application is typical of a commercial client/server application where many small transactions are being continuously processed. It is not representative of a decision support workload nor that of a file server workload, which are the other two components of the SCB. Additional information on file server performance can be found in Chapter 9, “AS/400 File Serving Performance” .

***SCB with Windows 3.1/95/NT Client***:

The specifications for the Server and Client listed below were the same for both the V5R1 and V4R5 peformance measurements:

***Measurement Configuration***:

AS/400 170 - 2292 - dedicated
1GB Memory
9-8GB Disk Drives (72 GB)
16Mbps Token Ring Lan - 2724 IOP

Clients:

- Win NT client - Pentium-133 MHZ - 32MB memory - Windows NT 4.0 - 1.2GB IDE Disk Drive
  Client Access Express for Windows

- Win NT client - Pentium-133 MHZ - 32MB memory - Windows NT 4.0 - 1.2GB IDE Disk Drive
  Client Access for Windows 95/NT V3R1M2

- 16 Mbps TR-LAN
  SCB Transaction-based Workload - implemented with Visual C++
  One client - Average transaction response time - 100 transactions
  (All response times are expressed in seconds)

**Client/Server Query and Decision Support**

Query and Decision Support (also known as ad-hoc queries) are database operations typically done throughout the day in many businesses. These operations are longer-running server-intensive database operations which are usually read-only. Unlike OLTP, they are seldom done frequently and although

throughput may not be a critical factor in performance, response time surely is. These queries usually result in minimal interactions between the server and the client. Although they may return a large number of rows of information, typically ad-hoc queries return few rows. Because these queries are server-intensive, they are good candidates for database serving. The response time of these remote (c/s) queries are not significantly slower (typically 5% - 30%) than local (host terminal) queries. However, since the server performance load can be large, the user may benefit from moving the execution of these queries off-shift.

In cases where the query executed is CPU-intensive and very few records are returned to the client, the response time of the query is typically very close to running the same query interactively on the server.

### ODBC Query Using Re-use Pre-started Jobs

V3R7 OS/400 provides a new option to re-use pre-started jobs for the QSERVER subsystem. This allows pre-started jobs to "recycle" jobs that have previously ended. As a result, CPU time to start-up jobs is reduced with the re-use value set to greater than 1.

This value can be changed with the AS/400 command CHGPJE. Use the QSERVER subsystem with the job QZDAINIT and subsystem QIWS for APPC jobs or QZDASOINIT for TCP/IP or IPX jobs. Press F10 for additional parameters then page down to the parameter "MAXIMUM NUMBER OF USES". Enter the new value for the maximum job re-uses. The V3R7 default value is 200.

## Query Download/Upload (Database file transfer)

Download/Upload queries represent a set of queries that either fetch a significant number of rows from DB2/400 tables (or files) or insert a significant number of rows into DB2/400 tables or files. Because of the number of rows processed, there is a significant amount of processing that occurs on the client. Many times, significant performance gains may be realized by running these types of queries on a faster client processor.

### Query Download Comparisons

This section compares the performance of downloading a significant number of records from an DB2/400 database file into a client application using various CA/400 APIs. To obtain the download comparisons, a client application was developed that fetches about 1.4 MBs of data using different APIs. All fetched character data was converted from EBCDIC to ASCII automatically by CA/400 functions. No further processing was performed on the data retrieved.

### Measurement Configuration:

The following configuration was used to perform the query download measurements:

```
AS/400 Server 50S-2120 - dedicated
256 MB Memory
2-6606, 2-6605 DASD (5.99 GB)
ValuePoint clients - '486-66 MHZ - 32MB memory - CA/400 for Windows 3.1 V3R1M1
2048 byte frame size
16 Mbps TR-LAN
```

The download operations were done using standard PC-based query tools. The tool was changed to display the first resulting rows or to simply indicate completion of the test. For query download tests, the ODBC.INI file on the client was changed to vary the Record Blocking size setting. The following queries were performed to download records from the DB2/400 database files:

*Table 10.1. CA/400 Windows 3.1 Query Download Tests*

| Query | SQL Statement | Row Size | Columns Per Row | Rows Fetched | Bytes Fetched |
|---|---|---|---|---|---|
| LBR | SELECT * FROM DBITRK/LBRSTATS | 118 | 10 | 12,000 | 1,416,000 |
| TRK | SELECT * FROM DBITRK/TRKOPRNS WHERE TRKITEMN <'ITE02210MNUMBR' | 314 | 42 | 4,418 | 1,387,252 |

**Measurement Results**:

The following table shows the ODBC.INI Record Blocking size setting, the overall download rates in megabytes per hour, the overall response times in seconds, and the AS/400 CPU seconds consumed for the queries listed above. The queries were implemented using Client Access/400 ODBC.

*Table 10.2. CA/400 Windows 3.1 Query Download Performance*

| Query | Record Block Size | Transfer Rate (MB/hr) | Response Time (seconds) | AS/400 CPU Consumed (seconds) |
|---|---|---|---|---|
| LBR - display rows | 512K | 296 | 17.2 | 0.63 |
| LBR - no display | 512K | 566 | 9.0 | 0.63 |
| LBR - no display | 32K | 520 | 9.8 | 0.69 |
| TRK - display rows | 512K | 271 | 18.4 | 0.78 |
| TRK - no display | 512K | 450 | 11.1 | 0.78 |
| TRK - no display | 32K | 427 | 11.7 | 0.86 |
| **Note**: All tests received all rows; "display" tests only displayed the first set of rows | | | | |

Query tools are available to provide the client with an easy, graphical way to access server databases. For performance reasons, many of these tools allow the user to limit the number of rows received. We ran one of the above queries and limited the rows using a query tool. This table shows the row limit, the ODBC.INI Record Blocking setting, and the overall response times in seconds.

*Table 10.3. CA/400 Windows 3.1 Query Download Performance - Limited Rows*

| Query | Row Limit | Record Block Size | Response Time (seconds) |
|---|---|---|---|
| LBR | 20 | 32K | 1.4 |
| LBR | 20 | 512K | 3.3 |
| TRK | 20 | 32K | 2.2 |
| TRK | 20 | 512K | 4.1 |
| **Note**: These tests only downloaded the first 20 rows of data | | | |

For comparison, we measured an IFS file transfer (download) operation using one of the same database tables above. We converted the DBITRK/LBRSTATS table to a stream file and loaded it on the server in the IFS root directory. We then used Windows File Manager to copy the file from the server to the PC hard disk. We compared the time for downloading a new file with replacing an existing file. The following table shows the overall download rates in megabytes per hour, the overall response times in seconds, and the AS/400 CPU seconds consumed for the queries listed above.

*Table 10.4. CA/400 Windows 3.1 File Download Performance*

| File Downloaded | File Transfer Rate (MB/hr) | File Transfer Time (seconds) | AS/400 CPU Consumed (seconds) |
|---|---|---|---|
| DBITRK/LBRSTATS - new file | 593 | 8.6 | 0.59 |
| DBITRK/LBRSTATS - replace file | 614 | 8.3 | 0.51 |

**Note**: These tests downloaded the file to the PC hard disk

**Conclusions/Recommendations**:

1. Only a small percentage of the total transfer time was due to the AS/400 CPU. Most of the time spent for large record download operations is in the client and communications time. Use fast clients for the best performance. Use fast communications adapters for higher throughput.

2. ODBC query download rates can be comparable to IFS file transfer rates.

3. For fastest retrieval times for an entire large database table, do not immediately format and display all the data retrieved. Instead, use client tools to manipulate and display the data after it has been entirely downloaded to the client.

4. When retrieving the entire database table, the recommended ODBC Record Blocking setting is 512 KB. Decreasing this size may cause slower performance. Memory-constrained clients may require a smaller block setting.

5. When using client tools to browse through the data, limit the query to display only the first screen of data. Fetch the next set of data when needed. Set the Record Blocking to 32K or less for fast retrieval of only a small number of rows from a large table.

6. As the number of columns to be retrieved increases, the retrieval rate decreases and response time increases.

7. The token-ring frame size used was 2K. Larger frame size settings may improve performance.

**Query Upload Scenario**

This section compares the performance of uploading a significant number of records into a DB2/400 database file from a Windows 3.1 application using various CA/400 ODBC APIs.

**Measurement Configuration**:

The following configuration was used to perform the query upload measurements:

```
AS/400 Server 50S-2120 - dedicated
256 MB Memory
2-6606, 2-6605 DASD (5.99 GB)
ValuePoint clients - '486-66 MHZ - 32MB memory - CA/400 for Windows 3.1 V3R1M1
2048 byte frame size
16 Mbps TR-LAN
```

The client application is written in C and utilizes CA/400 Windows 3.1 ODBC APIs to do single inserts and blocked inserts to a table within the DB2/400 database.

The following table gives a brief description of the SQL statements issued and the row descriptions. Note that the questions marks ("?") within the statements are parameter markers or variables that the client application supplies to the ODBC APIs.

| *Table 10.5. CA/400 Windows 3.1 Query Upload Tests* | | | | |
|---|---|---|---|---|
| **SQL Statement** | **Row Size** | **Columns Per Row** | **Rows Inserted** | **Bytes Inserted** |
| INSERT INTO JMBCOLL.PERF VALUES (?,?,?,?,?,?,?,?,?,?) | 100 | 10 | 1,000 | 100,000 |
| INSERT INTO JMBCOLL.PERF 350 ROWS VALUES (?,?,?,?,?,?,?,?,?,?) | 100 | 10 | 30,450 | 3,045,000 |

**Measurement Results**:

The following table shows the overall upload rates in megabytes per hour for the above SQL statements. The single insert case sends 1000 ODBC SQLExecute commands to the AS/400 server to perform 1000 inserts while the blocked insert scenario sends 87 ODBC SQLExecute commands to the server to perform 30,450 inserts.

| *Table 10.6. CA/400 Windows 3.1 Query Upload Performance* | | |
|---|---|---|
| **SQL Insert Row Count** | **Rows Inserted** | **Insert Rate (MB/hr)** |
| 1 | 1,000 | 16 |
| Block of 350 | 30,450 | 215 |

**Conclusions/Recommendations**:

1. Use blocked insert when possible

   Client applications that perform inserts, updates or deletes will generally perform these SQL commands one at a time to the CA/400 data access server. However, for inserts, there is an opportunity to use the blocked INSERT SQL statement which can be used to send a set of rows to the server in a single communications flow. Measurements have demonstrated that this form of insert can be over 20X faster than doing inserts one at a time.

2. Use faster clients

   A large portion of upload and download operations is due to the client. Increasing the speed of the client can improve throughput.

3. Use faster communications adapters

   Using slower communications adapters can result in costly delays. Upgrading the communications adapters can improve throughput.

## Client Access/400 for Windows 95/NT 5250 Emulator Performance

This section shows the performance of 5250 emulation using Client Access for Windows 95/NT compared to a 5250 terminal.

Client Access provides the capability to emulate 5250 terminal sessions with the flexibility to configure the keyboard and display to the user's preferences.

**5250 Emulator Performance Results**

**Measurement Configuration**:

AS/400 200-2030 - dedicated
V3R2
16 MB Memory - 4-6609 DASD (8 GB)
Clients:

- ValuePoint client - '486-66 Mhz - 32MB memory - Windows 95
- Pentium client - Pentium-133 Mhz - 32MB memory - Windows 95

Compared to 5250 display: 5291 model 2
CA/400 for Windows 95/NT V3R1M2
16 Mbps TR-LAN
Typical OS/400 5250 workstation screens - 24x80 resolution

**Measurement Results**:

*Table 10.7. CA/400 Windows 95/NT 5250 Emulator Performance*

**5250 Emulator Performance Comparison**
**CA/400 for Windows 95/NT V3R1M2**
**AS/400 - V3R2 200-2030**
**Client - '486 @ 66 MHZ versus Pentium @ 133 MHZ versus 5250 Display**

| Workstation Screen | Win-95 '486 Resp Time (seconds) | Win-95 Pentium Resp Time (seconds) | NT Pentium Resp Time (seconds) | 5250 Display Resp Time (seconds) |
|---|---|---|---|---|
| WRKACTJOB | 0.86 | 0.79 | 0.72 | 0.63 |
| WRKLIB *ALL | 0.80 | 0.70 | 0.69 | 0.59 |
| WRKOBJ *FILE | 0.63 | 0.55 | 0.46 | 0.38 |

**Note**: Average response time (seconds)

**Conclusions/Explanations**:

1. Client Access/400 for Windows 95/NT provides good 5250 emulator performance. NT provides slightly faster performance than Windows-95.

2. Faster clients provide faster response time. The Pentium times provide response times closer to the 5250 display response times.

## 10.4 Tips for Improving C/S Performance

Following are some tips to use when writing a client/server application which will provide the best performance:

1. Choose the right client processor.

   In many queries the majority of response time is due to client processing, especially when utilizing client/server tools (eg. 4GL/CASE tools). If response time is critical, choose a fast client

processor. Figure 28 on page 0 shows the potential increase in response times for queries relative to performing the queries on a '486 @ 66MHZ client:

2. Ensure that the client is optimized for performance

   Client/server applications usually require a large amount of processing power for both the client and the server. The client processor speed must be appropriate for the task. Also, memory requirements for the client may be large (toolkits, communications -- routers, buffers, etc., and operating system requirements). The response time variation between a fast well-tuned client and a under-powered client can be astounding.

3. Use the fastest communications media possible and tune the communications configuration settings

   Most client/server applications tend to send a large number of requests and responses between the client application and the server. To minimize the delay due to this communications traffic, use fast media such as local area networks (LANs) to attach clients to the server. If response times are critical, do not use wide-area networks (WANs) since the communications speeds are typically measured in thousands of bits per second instead of millions of bits per second. Also, be wary of bridges, routers, and gateways since they may introduce delays when communicating across networks. Instead, keep the response time-critical clients on the same network as the server.

   Use a fast PC communications adapter -- especially for file transfer operations. The communications adapter can be a major factor in constrained throughput. For download operations, a slow communications adapter can reduce throughput by over 10X compared to a fast adapter. The adapter can not keep up with the server and this results in overruns. When an overrun occurs, the server must detect the error and resend the data. This can result in large delays. If a faster adapter can not be used, communications can be tuned to reduce overruns. Following are some examples:

   - TCP/IP for Windows 95/NT

     When using the native Windows 95 TCP/IP communications stack, a registry entry can be changed to improve problems when using slow adapters or slow PCs. This setting can reduce performance on fast adapters and should only be changed when client adapter problems exist.

     "HKEY_LOCAL_MACHINE"

     Use REGEDIT to add a new string value "DefaultRcvWindow". Set the value to 4096 or decrease until retries are reduced.

     When using Windows NT, the value to change is: "HKEY_LOCAL_MACHINE"

     Use REGEDT32 to add the new REG_DWORD value "TcpWindowSize". Set the value to 4096 or decrease until retries are reduced.

   - APPC

     Set the LANWDWSTP setting from the default of 0 to 2 or greater. For slow adapters, this

will reduce the time to correct data re-transmission problems.

- Netsoft APPC Router

  Consider increasing the parameter MAXDATA size from the default value of 521 to the maximum size. This value is specific to each router and can be different for each router configured. The MAXDATA size must be equal to or less than the frame size opened for the network adaptor. Increasing this value can improve performance, in particular the performance of large data transfers.

  To change this parameter, open the 'Netsoft Administrator' folder, select 'Set Properties' of specific AS/400 configuration. Next select 'Properties' of the link being used (for instance 802.2). Finally, select the 'Advanced' tab.

- IPX/SPX

  If you are using IPX/SPX for large file transfers the default data size sent to IPX/SPX may be increased. Create the string value:

  "HKEY_CURRENT_USERAccessInternal_Components(Your Env)(Your System)IPX Max Send

  The default is 1400, the maximum is 65536. Setting this value above the default may cause errors in some configurations. If problems appear after changing this value, delete the registry entry.

The frame size and buffer size for your network card should be increased to optimize the network traffic for your situation. Large file transfers perform better with larger frame sizes if your network adapter and network devices support the larger sizes. Increased buffers allow the client to offload more work to the network card. These settings can usually be controlled through the Control Panel/Network/Adapter Properties, check with your network adapter manufacturer for details. Increasing these setting will increase the system resources used by your network adapter.

Consider the following tuning tips for AS/400 communication.

- Consider increasing the Maximum Transmission Unit (MTU) from the default value of 576. The AS/400 defaults to 576 when a route is added to the configuration (via CFGTCP option 3). This value ensures packets will not be dropped over this route as all TCP/IP implementations have to support at least a 576 byte Transmission Unit.

  In many cases, this value is unnecessarily small. For instance, if the route will only be used on the configured Ethernet or Token Ring, and there are no intermediate hops that only support a 576 byte packet. If this is the case, change the Route Maximum Transmission Unit size to *IFC. This will change the MTU on the Route to the Interface MTU size which defaults to the Line Description Frame size. This defaults to approximately 2000 for Token Ring and 1500 for Ethernet.

- Consider increasing the TCP receive buffer size from the default size of 8192 bytes to a larger value, for example 64384 bytes (via CFGTCP option 3).

This value specifies the amount of data the remote system can send before being read by the local application. If there are many retransmissions occurring due to the overrunning of a network adapter, decreasing this value instead of increasing it, could help performance.

- Consider increasing the TCP send buffer size from the default size of 8192 bytes to a larger value, for example 64384 bytes (via CFGTCP option 3).

  This value provides a limit on the number of outgoing bytes that are buffered by TCP. If there are many retransmissions occurring due to the overrunning of a network adapter, decreasing this value instead of increasing it, could help performance.

- Refer to section Chapter 5, “Communications Performance” on page 0 for AS/400 communication tuning guidelines and specifically Section 5.2, “LAN Protocols, Lines, and IOPs” on page 0.

ANYNET support allows clients to run APPC based applications over TCP/IP. ANYNET can be considerably slower than TCP/IP and consumes more CPU than TCP/IP. Client Access for Windows 95/NT allows clients to access the AS/400 directly through TCP/IP. TCP/IP provides faster response times than ANYNET.

4. Ensure that all database requests are optimized for performance

   Although the AS/400 database manager does a good job of handling database requests, it is important that performance-sensitive operations be tuned for optimal performance. Examples of database tuning are: ensure that indexes are being used, simplify SQL statements, minimize redundant operations (re-preparing SQL statements, etc.), and reduce the number of communications requests/responses by blocking. Use tools such as communications trace (STRCMNTRC), DB2/400 debug (STRDBG), Explain function (PRTSQLINF), and Performance Monitor (STRPFRMON) to assist with performance tuning.

5. Ensure that the database access method is tuned for performance

   Whether the application uses DRDA, DDM, Remote SQL, DAL, JDBC, or ODBC, tuning the access method can result in performance improvement. Client Access/400 ODBC support allows each client to customize the ODBC interface to the AS/400. This is done using the ODBC.INI file for Windows 3.1 or the ODBC Adminstrator for Windows 95/NT. For ODBC tuning guidelines, see section "ODBC Performance Settings" on page 144.

6. If a CASE/4GL toolkit is used, tune the application for performance

   Client toolkits can provide large improvements in application development productivity. But, since most are developed to communicate with multiple servers, they may not be optimized for any specific server. For more information on CASE/4GL toolkits, see section"Client/Server 4GL and Middleware" .

7. Use parameter markers support when performing repetitive transactions

   A parameter marker is a question mark (?) that appears in the SQL statement where a host variable could appear if the statement API string was a static SQL statement. Parameter markers enhance performance by allowing a user to prepare a statement once and then execute it many times using a different set of values for the parameter markers.

8. Reuse prepared statements

   Prepares of SQL statements can take asignificant amount of time. There are two ways to reuse prepared statements:

- Only prepare statements once (using parameter markers) and use SQLExecute ODBC API

  Reducing redundant prepare statements and using parameter markers instead of literals are two of the best ways to improve database server performance -- especially OLTP operations which are frequently repeated. Response time of a complex, repetitive transaction can be reduced by over 5X by changing the client application to take advantage of these improvements.

- Use package support

  Package support, available with CA/400 ODBC, provides built-in reuse of prepared statements. See "ODBC Performance Settings" on page 144 for more information on configuring for package support.

9. Use stored procedures and triggers to reduce communication flows

   To reduce network traffic between the client and the server and reduce response time, use stored procedures and/or triggers. Typical database serving applications send or receive from a dozen to a hundred requests/responses. Stored Procedures and triggers can reduce the number of flows significantly. Also, more processing is done at the server so the application can be completed more efficiently.

10. When possible, use SQLExecDirect for one-time execution (one flow, not two)

    SQLExecDirect can replace the pair: SQLPrepare and SQLExecute. **However**, if you are doing multiple executions of the SQLStatement (looping), you should separate the SQLPrepare and SQLExecute such that the SQLPrepare is done only once and the SQLExecute is processed multiple times. This reduces both AS/400 and client processing time because the PREPARE/DESCRIBE steps do not need to be repeated. This is much more efficient than SQLExecDirect.

11. Ensure that each statement has a unique statement handle

    Sharing statement handles for multiple sequential SQL statements causes DB2/400 to do FULL OPEN operations since the database cursor can not be re-used. 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 being executed in a loop. Allowing each SQL statement to have its own handle reduces the DB2/400 overhead.

12. Utilize blocking

    - Use "FOR FETCH ONLY" and avoid "UPDATE WHERE CURRENT OF"

    - Set maximum frame size > 2K for large upload or download

      For the Windows 3.1 client, use the Global Options settings in Configuration to set the maximum frame size. For the Extended DOS client, use the TRMF setting in CONFIG.PCS.

    - Use blocked inserts

      Blocked Insert allows a client application to send a set of rows to the server (instead of one at a time). Measurements show that the performance of Blocked Insert can exceed 10X improvement over single row insert (eg. 1000 100-byte rows inserted)

13. Use lowest level of commitment control required

    More server processing is required to process more stringent commitment control settings.

14. Define client column parameter marker variables identical to host column descriptions to allow for direct mapping on the server.

    This reduces the overhead of variable type mapping.

15. Consider tuning some CASE/4GL applications (changing ODBC APIs)

    Customizing "open" client applications by using the tips listed above, you may be able to improve overall performance.

16. Choose a server access method which provides high performance database serving

    If your 4GL supports multiple access methods to the AS/400 server, consider the following:

    a. Use ODBC for best SQL access performance

       ODBC can improve performance over other SQL access methods. ODBC is the strategic database serving interface to AS/400.

    b. DRDA

       Distributed Relational Database Architecture (DRDA) provides acceptable performance in most cases. When possible, use static SQL statements for the best performance.

    c. DDM

       Distributed Data Management (DDM) does not have the flexibility of SQL but, in most cases, provides good record-level file access performance.

    d. JDBC

Java Toolbox provides good C/S performance for client Java Aplications

17. Use client tools to assist in tuning the client application and middleware. Tools such as ODBCSpy and ODBC Trace (available through the ODBC Driver Manager) are useful in understanding what ODBC calls are being made and what activity is taking place as a result. Client application profilers may also be useful in tuning client applications and are often available with application development toolkits.

18. When possible, avoid extra communications layers such as AnyNet for the best performance of OLTP and large record upload/download workloads. Functions that do not require fast response times through the communications layers (e.g., ad-hoc queries and stored procedures) are a better fit for Anynet.

**ODBC Performance Settings**

You may be able to further improve the performance of your ODBC application by editing the ODBC.INI file on Windows 3.1. The settings in the ODBC.INI file are stored in the registry for Windows 95/NT. The recommended way to access these settings is through the ODBC Administrator in the Control Panel. The settings can be found in the registry under the Key "HKEY_CURRENT_USER.INI". The ODBC.INI file for Windows 3.1 clients contains information relating to the various ODBC drivers and data sources and is located in the Windows subdirectory for each CA/400 ODBC client. Listed below are some of the parameters that you can set to better tune the performance of the Client Access/400 ODBC Driver. The ODBC.INI performance parameters that we will be discussing are:

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

***Prefetch = choices 0, 1):*** 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. This option is set OFF by default. We recommend that this setting be turned ON. However, if the client application uses EXTENDED FETCH (SQLExtendedFetch) this option should be turned OFF.

***ExtendedDynamic = (choices 0,1):*** Extended dynamic support provides a means to "cache" dynamic SQL statements on the AS/400 server. With extended dynamic, information about the SQL statement is saved away in an SQL package object on the AS/400 server the first time the statement is run. On subsequent uses of the statement, CA/400 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 keyword. If the value for this keyword is 0, no packages are used and no additional information will be added to the ODBC.INI file. If the value is set to 1 (default), when an application is run for the first time, the ODBC driver will add a line to the ODBC.INI file for the datasource in use that looks like this:

```
Package<Appname> = lib/packagename,usage,pkg full option,pkg not used option
```

Once this entry is added to the ODBC.INI file it can be modified to provide the support that the user wants.

Packages may be shared by several clients to reduce the number of packages on the AS/400 server. For the clients to share the same package, 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 AS/400 and the clients given different libraries to store the package in. The location of the package is stored in the ODBC.INI file for Windows 3.1 and in the registry for Windows 95/NT.

***Usage (choices 0,1,2):*** The default and preferred performance setting (2) 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 or by changing the ODBC.INI file.

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. For each user to have a private package, the ODBC.INI file must be modified so that each user has a different package name. Either the library name or all but the last 3 characters of the package name can be changed.

***RecordBlocking = (choices 0,1,2):*** The RecordBlocking switch allows users to control the conditions under which the driver will retrieve multiple rows (block data) from the AS/400. The default and preferred performance setting (2) will enable blocking for everything except SELECT statements containing an explicit "FOR UPDATE OF" clause.

***BlockSizeKB = (choices 1 thru 512):*** The BlockSizeKB parameter allows users to control the number of rows fetched from the AS/400 per communications flow (send/receive pair). This value represents the client buffer size in kilobytes (1kb=1024) and is divided by the size of one row of data to determine the number of rows to fetch from the AS/400 server 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 = (choices 0,1):*** The LazyClose switch allows users to control the way SQLClose commands are handled by the Client Access/400 ODBC Driver. The default and preferred performance setting (1) enables Lazy Close. Enabling LazyClose will delay sending an SQLClose command to the AS/400 until the next ODBC request is sent. If Lazy Close is disabled, a SQLClose command will cause an immediate explicit flow to the AS/400 to perform the close. This option is used to reduce flows to the AS/400, and is purely a performance enhancing option.

***LibraryView = (choices 0,1):*** The LibraryView switch allows users to control the way Client Access/400 ODBC Driver deals with certain catalog requests that ask for all of the tables on the system. The default and preferred performance setting (0) 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 1 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.

## AS/400 Memory Requirements

Multiple clients running the CPW workload were used to help determine the optimal amount of AS/400 memory needed per client. For V4R2, It was found that in the range of 2.5 to 2.8 MB per user, client response times "leveled off" such that more memory did not significantly improve response times. However, as we continued to add memory to the pool beyond 2.8 MB per user, the paging and faulting for that pool continue to decrease significantly until about 3.2 MB per user was available for each client. See "Client/Server Online Transaction Processing (OLTP)" for more information on the CPW workload.

The effects of memory depends on the kind of workload done by the jobs in the shared pool. Your memory requirements may vary depending on many factors such as high communications I/O, accessing very large database tables, frequent DASD I/O accesses, and high level of multi-processing (sharing) in the pool.

Note that the AS/400 Work Management manual (SC41-3306) has a good set of recommendations on memory tuning. Knowledge of these tips can assist in building high-performance system environments.

# Chapter 11. Domino for iSeries

This section includes performance information for Lotus Domino for AS/400. Domino for AS/400 provides many functions. However, this section focuses on the performance of the mail server function.

Many factors impact overall performance (e.g., end-user response time, throughput) in the iSeries 400 Domino environment, some of which are listed below:

- iSeries 400 processor speed
- Utilization of key iSeries 400 resources (CPU, IOP, memory, disk)
- Object contention (e.g. mutex waits, lock waits)
- Speed of the communications links
- Congestion of network resources
- Processing speed of the client system

The primary focus of this section will be to discuss the performance characteristics of the iSeries 400 as a server in a Domino environment, providing capacity planning information and recommendations for best performance.

**V5R1 Updates** - please look for new information in this chapter on the following topics:
- V5R1 Server response time improvements
- Domino Performance Statistics
- Five new Dedicated Server for Domino (DSD) offerings
- Significantly enhanced processing capability for DSD models called complementary processing

In general, OS/400 V5R1 delivers performance equivalent to V4R5 for Domino environments for existing systems. The new V5R1 270 and 8xx systems deliver significantly higher capacities for Domino processing on DSD models as well as for traditional models. See Section 11.12 for information on Domino mail and calendaring performance for the new V5R1 models.

## 11.1 Workload Descriptions

The mail and calendaring workload and the simple mail workload scenarios were driven by an automated environment which ran a script similar to the mail workload from Lotus NotesBench. Lotus NotesBench is a collection of benchmarks, or workloads, for evaluating the performance of Domino servers. The results shown here are not official NotesBench measurements or results. The numbers discussed here may not be used officially or publicly to compare to NotesBench results published for other Notes server environments. For official iSeries audited NotesBench results, see http://www.notesbench.org . (Note: in order to access the NotesBench results you will need to apply for a userid/password through the Notesbench organization. Click on Registration at the above address.)

- **Mail and Calendaring Users (MCU)**
  Each user completes the following actions an average of every 15 minutes except where noted:

  - Open mail database which contains documents that are 10Kbytes in size.
  - Open the current view
  - Open 5 documents in the mail file
  - Categorize 2 of the documents

- ❖ Compose 2 new mail memos/replies 10Kbytes in size. (every 90 minutes)
- ❖ Mark several documents for deletion
- ❖ Delete documents marked for deletion
- ❖ Create 1 appointment (every 90 minutes)
- ❖ Schedule 1 meeting invitation (every 90 minutes)
- ❖ Close the view

- **Simple Mail Users (SMU)** (this workload will not be used in V4R5 and beyond)
  Each user completes the following actions an average of every 15 minutes except where noted:

  - ❖ Open mail database which contains documents that are 1 KB in size.
  - ❖ Open the current view
  - ❖ Open 5 documents in the mail file
  - ❖ Categorize 2 of the documents
  - ❖ Compose 2 new mail memos/replies 1 KB in size(every 90 minutes)
  - ❖ Mark several documents for deletion
  - ❖ Delete documents marked for deletion
  - ❖ Close the view

- **Web mail**

  The web mail workload scenario is very similar to the Simple Mail workload except that the Domino mail files are accessed through HTTP from a Web browser. There is no scheduling or calendaring taking place. The workload script issues an HTTP request approximately every 2 minutes. When accessing mail through Notes, the Notes client performs the majority of the work. When a web browser accesses mail from a Domino server, Domino bears the majority of the processing load because nothing is being run or stored on the web browser. The browser's main purpose is to display information.

---

## 11.2 Domino R5.0.

Release 5.0 of Domino (R5 for short), included a major initiative to improve Domino performance. Some of the changes include:

1. The ability to use a pool of server threads utilizing I/O completion ports (IOCP).
2. Redesign of the on disk structure (ODS) for storing databases.
3. Memory and I/O optimization
4. Ability to use multiple mail.box files.
5. Transaction logging to improve server recovery time and data integrity.
6. The ability to assign individual priorities to tasks within a server (see section 11.8).

These changes and others helped reduce the number of disk I/Os and the amount of CPU time used. R5 with IOCP (5.0.1) shows a marked improvement over 4.6 and R5 without IOCP (5.0). Figure 11.2 shows the processor cost per user as the number of users in a partition increases. R5 with IOCP has the best performance at all points. Note that from 1000 users to 5000 users, the processor cost was less than 10%, compared to 70+ percent for 4.6 and 25% for R5 with no IOCP.

The net result of all this is that if you have many partitions with low number of users, for a nominal cpu cost, you now have the option of combining those into a single partition (up to 7000 active mail users) .

This is a tradeoff between manageability and performance, because as you add users to a partition, the CPU cost per user increases (from 1000 to 5000 the cost goes up 10%).

However, if you have multiple mail servers that can be combined into 1 large server, the cost of mail routing may be reduced significantly if the mail no longer routed out of the server (mail is delivered locally within the single large server). Our tests showed a 10% reduction in total cpu by going from multiple servers to a single server. We also saw a reduction in memory requirements.

The following graph shows how the processor costs will increase as the number of users in a partition increases. This increase is due to management overhead and contention for Domino resources.

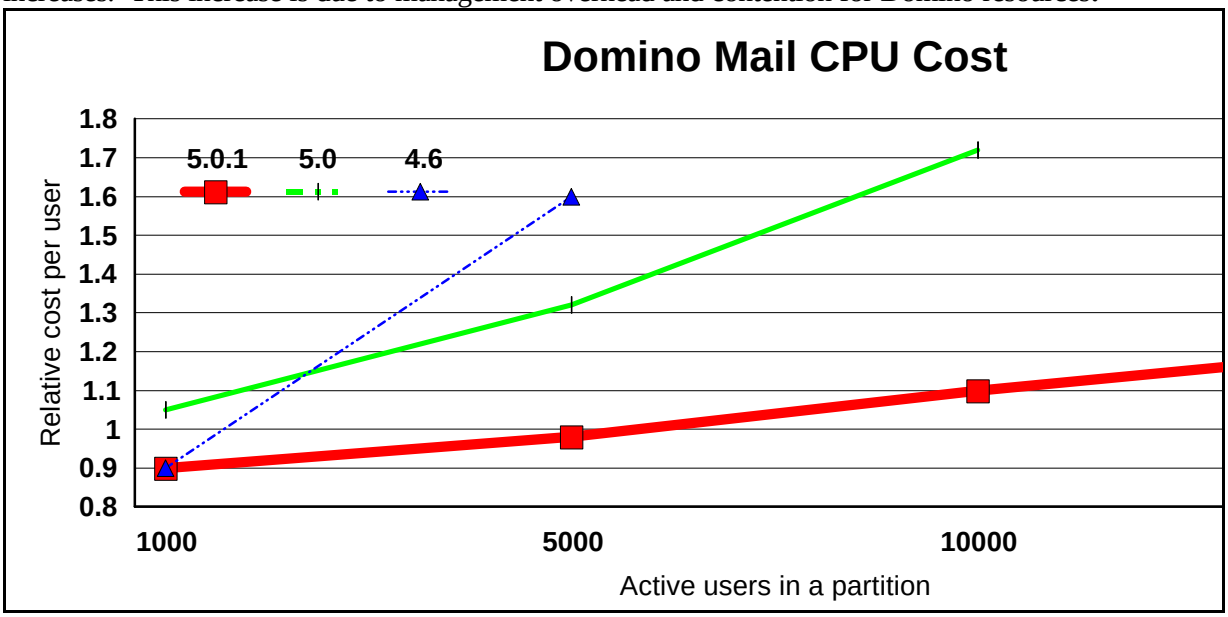



*Figure 11.2 CPU cost per Domino Mail User.*

Additional conclusions and recommendations that we can draw from the R5 data are as follows:

1. Starting with R5.0.1, Domino will default to using IOCP. You should remove the following notes.ini variables:
   - SERVER_MAXINITIALTHREADS
   - SERVER_SECONDARYTHREADS
2. We ran our tests on the smaller systems with the NSF_Buffer_Pool_Size_MB set to 300. This reduced the CPU utilization by a small percentage, but reduced the page faulting demands significantly when compared to measurements set to 507MB. On the larger systems, we ran with the NSF_Buffer_Pool_Size_MB set to 200, 300, 500, 600, 1000, and 1500. The general trend was the larger the buffer pool size, the higher the fault rate, but the lower the cpu cost. A good starting point is ¼ the size of the storage pool divided by the number of partitions (you can prorate the amount based on how many users are in each partition). If the faulting rate looks high, decrease the buffer pool size. If the faulting rate is low but your cpu utilization is high, try increasing the buffer pool size. Increasing the buffer pool size allocates larger objects specifically for Domino buffers, thus increasing storage pool contention and making less storage available for the paging/faulting of other objects on the system. To optimize performance, increase the buffer pool size until it starts to impact the faulting rate

then back it down just a little. Changes to the buffer pool size in the Notes.ini file will require the server to be restarted before taking effect.

3. The tests on the 840 24-way and some on the 820 4-way used 100MB Ethernet to connect the clients and servers in the LAN. The other measurements systems used Token Ring. As long as the media was not over utilized, end user response times were very similar between these two protocols. See data in Table 11.1.
4. Enabling transactional logging typically adds CPU cost and additional I/Os. These CPU and disk costs can be justified if transactional logging is determined to be necessary for server reliability and recovery speed. Data in Table 11.1 provides a comparison of 6000 Mail and Calendar Users with and without transactional logging active. For the tests with transactional logging active, the logs were placed in a separate ASP. This will typically provide better performance than having the logs on the disk same drives as the Domino databases.
5. While R5 caused an increase in disk utilization over 4.6, the actual number of disk reads and writes were reduced by 5%. The disk utilization increased because the disk write cache was being overwritten due to the increase in the size of the disk writes(10K vs 7K in our Simple mail workload) and the way the disk writes were not spread evenly over time. V4R4 will compensate for most of this with its IFS synch daemon redesign. For V4R3, you will need to ensure you have enough disk configured to keep the disk %busy below the recommended guidelines.
6. Running the simple mail user tests with R5 clients vs. 4.6 clients showed no performance difference.
7. The clients in the R5 tests were able to sign on 2.5 times as fast as the 4.6 users. This is due to the resource management improvements in Domino R5.
8. Specific tests to analyze the improvement of using multiple mail.box files showed a 1-2% improvement in overall cpu utilization. Knowledge of mail routing algorithms combined with the tests we ran suggest that 1 mail.box is sufficient for low number of users in a partition, 2 is sufficient for most environments, and more than 2 does not improve performance significantly, but still may be beneficial, up to 4 mail.boxes. On the 840 24-way, we ran with 10 mail.boxes on the two hubs, which is the maximum allowed.

## 11.3 V5R1 Response Time Improvements

The new V5R1 iSeries 400 model 270/8xx Servers are equipped with faster processors than previous iSeries 400 servers, and deliver faster response times with “equivalent” megahertz (sum total of MHz x number of SMP processors). Of course other factors besides CPU time need to be considered when evaluating overall performance, but for the CPU portion of the response time the following applies: faster megahertz processors will deliver better response times than an “equivalent” total amount of megahertz which is the sum of slower processors. For example, the 270-2423 processor is rated at 450mhz and the 170-2409 has 2 processors rated at 255mhz; the 450mhz processor will provide better response time than 510mhz (2 x 255mhz). Now, with V5R1, the 540mhz & 600mhz processors perform even faster. Figure 11.3 below depicts the response time performance for three processor types over a range of utilizations. Actual results will vary based on the type of workload being performed on the system.

Using a web shopping application, we measured the following results in the lab. In tests involving 100 web shopping users, the 170-2409 ran at 71.5% CPU utilization with .78 seconds average response time. The 270-2423 ran at 73.6% CPU with average response time of .63 seconds. This shows a response time improvement of approximately 20% near 70% CPU utilization which corresponds with the data shown in Figure 11.3. Response times at lower CPU utilizations will see even more improvement from faster processors. The 270-2454 was not measured with the web shopping application, but would provide even better response times than the 270-2423 as projected in Figure 11.3 below.

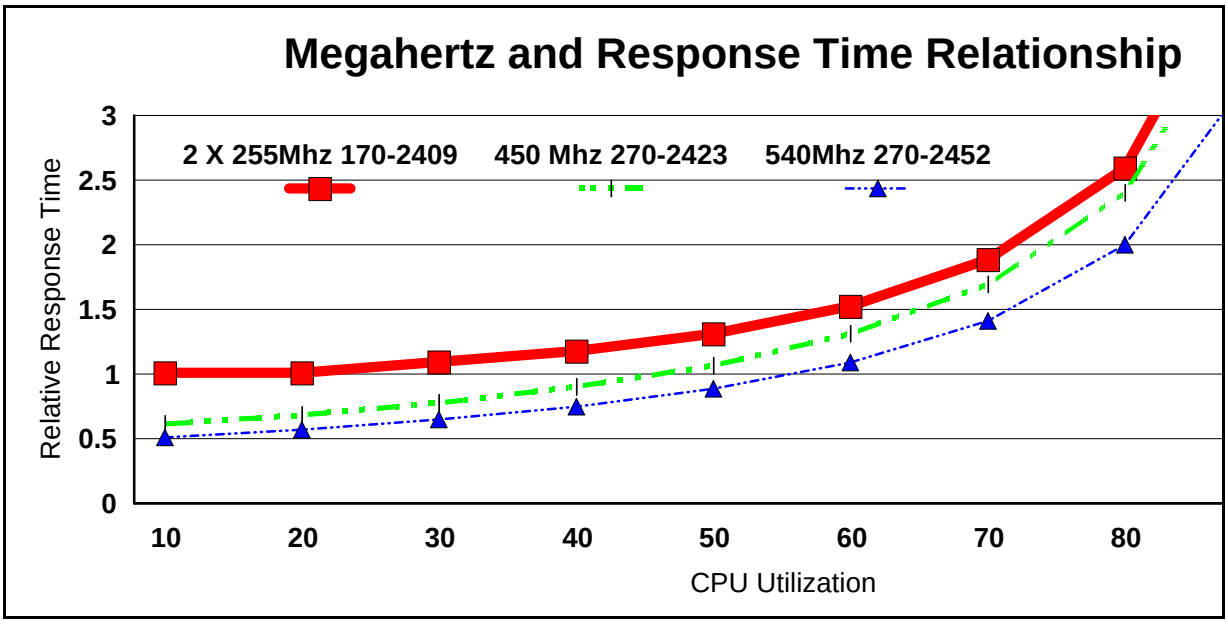



*Figure 11.3 Response Time and Megahertz relationship*

## 11.4 Dedicated Server for Domino

AS/400e Dedicated Server for Domino (DSD ) processor features deliver exceptional price performance for AS/400 environments running Domino. The first DSD servers were introduced 8/3/99 and now, with V5R1, the third generation of DSDs are being delivered. Five new V5R1 DSD features provide a wide range of performance and increased processing capability. There are two Model 270 features: 2452(540mhz Uni), and 2454(600mhz 2-way), and three Model 820 featues: 2456(600mhz Uni), 2457(600 MHz 2-way), and 2458(600mhz 4-way). Additional information and specifications, including Mail and Calendar User ratings, for DSD and traditional iSeries servers can be found in Appendix D, *iSeries CPW Values*.

With V5R1, the DSD has been enhanced to provide processing capacity for Domino-complementary processing such as from Java Servlets and WebSphere Application Server integration. Many workloads that were previously treated as "non-Domino processing" on the DSD will now be treated as "Domino processing" when used in conjunction with Domino. This enhanced behavior which supports Domino-complementary workloads on the DSD is available after September 28th, 2001, with the refreshed version of OS/400 V5R1. This enhanced behavior is applicable to all DSD models including the model 170, and the previous 270 and 820 models. Additional information on the Domino-complementary performance behavior can be found in Chapter 2, *AS/400 RISC Server Model Performance Behavior*, in section 2.13, *Dedicated Server for Domino Performance Behavior*. In addition, a white paper, *Enhanced V5R1 Processing Capability for the iSeries Dedicated Server for Domino*, provides expanded information on DSD performance behavior and can be accessed at:
**http://www.ibm.com/eserver/iseries/domino/pdf/dsdjavav5r1.pdf** .

Prior to this enhanced behavior, it was recommended to keep “non-Domino processing” below 10-15% of the system capacity, whether or not the function is used in conjunction with Domino. Now, in V5R1, workloads used in conjunction with Domino are treated as “Domino processing”, and only the processing in DB2 Universal Database access should be kept below 15% of the system capacity. If these same workloads are run on the DSD without Domino, then all processing is considered “non-Domino” and should be kept below 10-15% of the system capacity, which is similar to the previous guideline.

If the 15% DB2 guideline is exceeded by Domino or Domino-Complementary processing, the threads or jobs that are using DB2 resources may experienced increased response times. Other processing on the system will not be impacted. Similarly, if workloads run without Domino, which are then considered non-Domino processing, exceed 10-15% of system capacity, they may experience increased response times. In this case CFINT processing may also be observed.

Please refer to data in table 11.1 which shows data for test performed on some of the DSD models with the Mail and Calendar User workload.

Please refer to http://www-1.ibm.com/servers/eserver/iseries/domino/support for details on PTFs and QMU levels required for DSD models.

---

## 11.5 Performance Tips / Techniques

1. As the number of active users in a partition increases, contention for resources will cause an increase in CPU consumption for each user (see figure 11.2). Best results, - in terms of performance and stability, -were achieved when the number of active users was limited to 3000, even though up to 7,000 active users were tested.

2. Initial user connection places the heaviest load on the iSeries 400, requiring the largest amount of CPU, main storage, and disk resources. When sizing a system, ensure there is sufficient capacity. for this activity as well as the typical peaks of activity throughout the day. R5 reduces the contention during initial signon, but does not eliminate it.

3. For the system storage pool in which the Domino server and users run, you will need to configure approximately 60MB + 1MB per active user. If you have multiple partitions, you should configure approximately 16MB for each additional server. Note this does not include storage required for other system storage pools (e.g. *MACHINE, *SPOOL, etc.). Adding more main storage than what is recommended here will provide even better response times and will also provide capacity for future growth or workload peaks. See section 11.9. , Mail Serving Capacity Planning, for more details on estimating 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 mail servers run in.

4. Our mail tests show a 10-15% reduction in cpu utilization with the system value QPRCMLTTSK(Processor multi-tasking) set to 1. This allows the system to have two sets of task data ready to run for each physical processor. When one of the tasks has a cache miss, the processor can switch to the second task while the cache miss for the first task is serviced. With QPRCMLTTSK set to 0, the processor is essentially idle during a cache miss.

5. iSeries 400 notes.ini / server document R5 settings:

- Server_Max_Concurrent_Trans
  **DO NOT SET THIS TO -1** as was previously recommended in 4.6. With R5 beginning in 5.0.1, setting this to -1 will cause the server to create a very large number of server threads. Our tests had this set equal to ServerPoolTasks or higher. If you set this too low, response time will suffer.

- Mail.box setting.
  Setting the number of mail boxes to more than 1 may reduce contention and reduce the CPU utilization. Setting this to 2, 3, or 4 should be sufficient for most environments.

  Note: starting in R5, this is in the Server Configuration document

- Mail Delivery and Transfer Threads

  You can configure the following in the Server Configuration document:

  - Maximum delivery threads. These pull mail out of mail.box and place it in the users mail file. These threads tended to use more resources than the transfer threads, so we needed to configure twice as many of these so they would keep up.
  - Maximum Transfer threads. These move mail from one server's mail.box to another server's mail.box. In the peer-to-peer topology, at least 3 were needed. In the hub and spoke topology, only 1 was needed in each spoke since mail was transferred to only one location (the hub). 25 were configured for the hubs (one for each spoke).
  - Maximum concurrent transfer threads. This is the number of transfer threads from server 'A' to server 'B'. We set this to 1, which was sufficient in all our testing.

- NSF_Buffer_Pool_Size_MB
  This controls the size of the memory section used for buffering I/Os to and from disk storage. If you make this too small and more storage is needed, Domino will begin using its own memory management code which adds unnecessary overhead since OS/400 already is managing the virtual storage. If you make it too large, Domino will use the space inefficiently and will overrun the main storage pool and cause high faulting. We ran tests all the way up to 1500MB. At this size, the cpu was reduced by a percentage point or two, but the page faulting was twice as much as when the buffer pool was set to 1000 MB. This is acceptable only if you have enough disk drives to service the faults. If not, reduce the buffer pool size. Refer to the discussion of NSF_Buffer_Pool_Size_MB in section 11.2.

- Server_Pool_Tasks
  In the NOTES.INI file starting with 5.0.1, you can set the number of server threads in a partition. Our tests showed best results when this was set to 1-2% of the number of active threads. For example, with 3000 active users, the Server_Pool_Tasks was set to 60. Configuring extra threads will increase the thread management cost, and increase your overall cpu utilization up to 5%.

- Route at once
  In the Server Connection document, you can specify the number of normal-priority messages that accumulate before the server routes mail. For our large server runs, we set this to 20.

Overall, this decreased the cpu utilization by approximately 10% by allowing the router to deliver more messages when it makes a connection, rather than 1 message per connection.

- Hub-and-spoke topology versus peer-to-peer topology.
  We attempted the large server runs with both a peer-to-peer topology and a hub-and-spoke topology (see the Domino Administrators guide for more details on how to set this up). While the peer-to-peer funtioned well for up to 60,000 users, the hub-and-spoke topology had better performance beyond 60,000 users due to the reduced number of server to server connections (on the order of 50 versus 600) and the associated costs. A hub topology is also easier to manage, and is sometimes necessitated by the LAN or WAN configuration. Also, according to the Domino Administrators guide, the hub-and-spoke topology is more stable.

6. AS/400 environment variable settings.
   - Notes_SHARED_DPOOLSIZE. It is best to not set this and let it default to 12000000. This value controls how Domino memory management is done. You will incur more management overhead if this is set lower than 12000000. Do not make this variable larger than 12000000.

   - Notes_AS400_CONSOLE_ENTRIES set to 10,000 (the default). This is the size of the console file that displays the status messages when you enter the DSPDOMCSL or WRKDOMCSL commands. As this file grows, the response time for these two commands increases.

   For more detail on the above settings, see the Domino Server Administrator Guide.

7. Dedicate servers to a specific task
   This allows you to separate out groups of users. For example, you may want your mail delivered at a different priority than you want database accesses. This will reduce the contention between different types of users. Separate servers for different tasks are also recommended for high availability.

8. MIME format.
   For users accessing mail from both the Internet and Notes, store the messages in both Notes and MIME format. This offers the best performance during mail retrieval because a format conversion is not necessary. NOTE: This will take up extra disk space, so there is a trade-off of increased performance over disk space.

9. Full text indexes
   Consider whether to allow users to create full text indexes for their mail files, and avoid the use of them whenever possible. These indexes are expensive to maintain since they take up CPU processing time and disk space.

10. Replication.
    To improve replication performance, you may need to do the following:
    - Use selective replication
    - Replicate more often so there are fewer updates per replication
    - Schedule replications at off-peak hours
    - Set up replication groups based on replication priority. Set the replication priority to high, medium, or low to replicate databases of different priorities at different times.

11. Unread marks.
Select "Don't maintain unread marks" in the advanced properties section of Database properties if unread marks are not important. This can save a significant amount of cpu time on certain applications. Depending on the amount of changes being made to the database, not maintaining unread marks can have a significant improvement. Test results in the lab with a Web shopping applications have shown a cpu reduction of up to 20%. For mail, setting this in the NAB decreased the cpu cost by 1-2%. Setting this in all of the user's mail files showed a large memory and cpu reduction (on the order of 5-10% for both). However, unread marks is an often used feature for mail users, and should be disabled only after careful analysis of the tradeoff between the performance gain and loss of usability.

12. Don't overwrite free space
Select "Don't overwrite free space" in the advanced properties section of Database properties if system security can be maintained through other means, such as the default of PUBLIC *EXCLUDE for mail files. This can save on the order of 1-5% of cpu. Note you can set this for the mail.box files as well.

13. Full vs. Half duplex on Ethernet LAN.
Ensure the AS/400 and the Ethernet switches in the network are both RUNNING full duplex in order to achieve maximum performance. Very poor performance will result if either is running half duplex and the other is running full duplex. This seems rather obvious, but one or the other of these may be running half duplex if they are not both set to full duplex or they are not both set to auto-negotiate. It is usually best to use auto-negotiate. Just checking the settings is not sufficient, a LAN tester must be plugged into the network to verify full vs. half.

14. Additional references
The following web site contains additional Domino information and white paper resources.
See **http://www.iseries.ibm.com/developer/domino/** then click on performance.
Some of the information will be redundant to what is provided in this document:

---

## 11.6 Web Mail

Conclusions we can draw from R5 Web mail tests and the data in table 11.4 are as follows:

1. Specific tests were run on the V4R5 systems to analyze performance improvements. The results show that a 2-Way 270 can support twice as many users as the 170 2-way system, and the 270 response time is 40% less than the 170 response time.
2. Web mail has improved dramatically in R5 and is undergoing continual improvement, therefore, you should always apply the latest fixes to maintain optimum performance. As new server code was produced and loaded on our AS/400's, we saw changes in response time and CPU %. Just from moving from R5.02 to R5.02a we saw CPU % drop from 30.1% to 27% and response time drop from 388 ms to 340ms.
3. Optimum results were achieved when the number of threads used (configured in the server document), was slightly larger than the max seen by doing `SHOW STAT DOMINO.THREADS.ACTIVE.PEAK`. Our tests showed a slight improvement in cpu of about 3%. However, if the number of threads is set very large, the CPU% will be much higher as well. Also, startup and shutdown will be needlessly lengthened. Therefore, do not set the number of threads much larger than the peak.

4. The cache size setting is another tuning parameter that affects performance. This can be set in the Server Configuration Document under the Internet Protocols/DominoWeb Engine/Memory Caches tabs. Look at SH STAT DOMINO statistics to determine if your caches are configured optimally. For example, if the displacement rate is high, you may want to increase the cache. Try a value at least 25% larger.
5. We ran our tests with the NSF_Buffer_Pool_Size MB set to 300. See the previous discussion on setting and maintaining this parameter in Sections 11.2 and 11.5.
6. The Web mail workload we tested used about 3-4 times more CPU and 50% more memory per user than the mail and calendaring workload tested. For detailed capacity planning, Web mail is included in the Workload Estimator. For more details, see **Appendix B**, **AS/400 Sizing**, or the following URL: http://as400service.ibm.com/estimator
7. Other items that may be set in order for web mail to run at its top performance include setting the following Notes.ini variables on the System Under Test.
   - DominoAsynchronizeAgents=1
     *DominoAsychronizeAgents* will allow your server to run web-triggered agents in parallel. This is important because Domino, by default, will run agents triggered by web browsers one at a time. Configuring your Domino server to run agents in parallel may improve your application response time.
   - DominoAnalyzeFormulas=1
     *DominoAnalyzeFormulas* allows your server to enable caching of dynamic pages. By default, Domino will not allow any dynamic pages that contain formulas to be cached, since the resulting values could quickly become outdated. However, you can configure your Domino server to do intelligent caching based on the volatility of the formulas used.

---

## 11.7 Domino Subsystem Tuning

The objects needed for making subsystem changes to Domino are located in library QUSRNOTES and have the same name as the subsystem that the Domino servers run in. The objects you can change are:

- Class (timeslice, priority, etc.)
- Subsystem description (pool configuration)
- Job queue (max active)
- Job description

The system supplied defaults for these objects should enable Domino to run with optimal performance. However, if you want to ensure a specific server has better response time than another server, you could configure that server in its own partition and change the priority for that subsystem (change the class), and could also run that server in its own private pool (change the subsystem description).

You can create a class for each task in a Domino server. You would do this if, for example, you wanted mail serving (SERVER task) to run at a higher priority than mail routing (ROUTER task). To enable this level of priority setting, you need to do two steps:
1. Create the classes that you want your Domino tasks to use.
2. Modify the following IFS file '/QIBM/USERDATA/LOTUS/NOTES/DOMINO_CLASSES'. In that file, you can associate a class with a task within a given server.
3. Refer to the release notes in READAS4.NSF for details.

## 11.8 Performance Monitoring Statistics

Function to monitor performance statistics was added to Domino Release 5.0.3 for AS/400. Domino will track performance metrics of the operating system and output the results to the server. Type "show stat platform" at the server console to display them. This feature is disabled by default in R5.0.3 and later versions. You can enable it by setting the parameter: PLATFORM_STATISTICS_ENABLED=1 in the NOTES.INI file and restarting your server.

Informal testing in the lab has shown that the overhead of having statistics collection enabled is quite small and typically not even measurable. For additional information on these performance metrics, go to: **http://www.iseries.ibm.com/domino/qmr503.htm** and click on "Lotus Domino for AS/400 5.0.3 Release Notes" which is near the bottom of the page.

## 11.9 Sizing Domino on iSeries

To compare Domino processing capabilities for the various iSeries servers, you should use the new CIW metric. CIW is described in detail in Appendix A and CIW ratings for the V5R1 iSeries servers can be found in Appendix D.

For sizing Domino mail and application workloads, the recommended method is the IBM Workload Estimator for iSeries. You can access the Workload Estimator from the Domino on iSeries home page (select "sizing Information") or at this URL: ***http://as400service.ibm.com/estimator*** . IBM and Lotus Representatives can access it for download from the Server Sales Web site (Select Proposal Resources, then Tools Downloads). Business Partners can access it from PartnerInfo (Select All Servers, then AS/400, then Proposal Resources, then Tools Downloads).

The Workload Estimator is refreshed 3 to 4 times each year, and significant enhancements have been added for Domino workloads during 2001. The estimator's rich help text describes the enhancements in more detail, but here is a brief overview of the Domino enhancements provided with the April 2001, June 2001, and October 2001 refreshes:

- The estimator now supports projecting other workloads besides Domino on the DSD. The Estimator allows the combined effect of Domino, Java, Net.Commerce, WebSphere, and traditional workloads to be simultaneously estimated for a single server.
  **Note**: The Workload Estimator will require that Domino comprise at least 50% of the projected CPU utilization when projecting for a Dedicated Server.

- A projection of DB2 database processing is provided for Domino workloads as well as other workload types such as Java, WebSphere, and HTTP.

- New defaults and assumptions were added based on feedback from business partners and technical specialists. The new defaults and assumptions for items such as average size of mail databases, and concurrency, are intended to reflect as closely as possible typical customer usage of Domino.

- New DSD Calculations -- New with OS/400 V5R1, the Dedicated Server for Domino (DSD) models will support Domino Complementary workloads such as Java Servlets and WebSphere Application Server. This includes the five new DSD models announced April 2001, as well as previous DSD models. When specifying V5R1 for OS Version, the Estimator will allow other workloads in addition

to Domino on a DSD as long as the following requirements are met: Domino must be the majority of the projected workload, and the projected DB2 Universal Database utilization is less than the rated DB2 capacity for that model. The rated DB2 capacity is 15% of the CPU for DSD models. The new V5R1 function which supports complementary workloads for DSD models will be available after September 28, 2001 with the refreshed version of OS/400 V5R1. Prior to that date you may want to specify V4R5 for OS Version when projecting Domino workloads for previously installed DSDS systems.

- Domino.Doc support has been incorporated as an application type in the enhanced support for application sizing.

- Removal of Domino 4.6 Support -- Domino version 4.6 will no longer be supported by the Workload Estimator. All Domino version 4.6 workloads restored will be automatically converted to Domino version 5.

- Two new mail access types were added: iNotes Outlook and iNotes Web Access.

- Anti-Virus protection was added as a option on the mail workload definition screen.

- The WebMail client type was updated to reflect results with Domino 5.08.

- The minimum amount of memory required per partition was increased to better size configurations having one or a small numbers of users in a partition, such as is found in some application development environments.

- Application sizing capabilities have been significantly enhanced. Application rating can be specified with increased precision, and the application examples that were previously described in the help text have now been incorporated as selectable application types. New application types have also been added. When defining your application workload(s), you can choose from these IBM defined application types, or choose the custom type and define your own application rating. The IBM defined applications types now include:
  - QuickPlace Light Publishing
  - Document Library
  - Customer Relationship Management
  - Web Shopper
  - Web Shopper with DB2
  - Domino.Doc - Casual
  - Domino.Doc - Moderate
  - Domino.Doc - Heavy

Additional information on sizing Domino HTTP applications for AS/400 can be found at http://www-1.ibm.com/servers/eserver/iseries/domino/d4appsz.htm. Several sizing examples are provided that represent typical Web-enabled applications running on a Domino for AS/400 server. The examples show projected throughput rates for various iSeries servers. To observe transaction rates for a Domino sever you can use the “show stat domino” command and note the Domino.Requests.Per1hour, Domino.Requests.Per1min, and Domino.Requests.Per5min results. The applications described in these examples are included as IBM defined applications in the Workload Estimator.

For more information sizing and capacity planning information, see **Appendix B, iSeries and AS/400 Sizing.**

## 11.10 SMU, MCU, and Typical

During the past couple years the metrics of SMU (Simple Mail Users), MCU (Mail and Calendaring Users), and Typical have been used to describe the Domino mail capabilities of AS/400 and iSeries servers. This section will explain the relationship between these metrics. The metrics were created in addition to the NotesBench public benchmarks because publishing results generated by NotesBench workloads without having them officially audited is prohibited. Because it is not possible to audit results for every server configuration that we need to provide a Domino mail rating for, the SMU, MCU, and Typical metrics are used. The SMU and MCU Workloads are described in detail in section *11.1 Workload Descriptions*.

The SMU and MCU workloads provide a good basis for comparison of servers, but are lighter than what would expected for a 'typical' mail users Therefore, a Typical user was defined which is described as performing a workload that is three times heavier than a SMU, and twice as heavy as an MCU. This means that a given server that can support X number of ‘typical’ users can support 3X SMU, and 2X MCU. To translate these metrics into terminology used by the IBM Workload Estimator for iSeries for defining Domino mail workloads, the equivalent of a ‘typical’ user is a ‘moderate’ mail user.

When using the Workload Estimator to project mail capabilities, the factor of concurrency also needs to be considered. For example, the 820-2457 is rated at 6660 MCU (at 70% CPU), and the Workload Estimator will indicate that this system supports 3330 users at 100% concurrency. Thus, our 2X factor for MCU versus Typical. Now, if you were to specify 6660 Notes users in the Workload Estimator using the default 50% concurrency and the default settings which equal a moderate (typical) user, the Estimator will project a CPU utilization close to 70% for the 820-2457. And by changing the concurrency rating to 100%, the number of users projected at 70% is 3330, or one half of the MCU rating for this server.

## 11.11 Mail and Calendaring Test Data

The following tables provide a summary of measured performance data. 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.

*Table 11.1. Mail and Calendar Serving Performance Data*

**Mail and Calendar Serving With Domino on iSeries 400 Server Models**

| Model | # Active Notes Users | # Domino Partitions | Main Storage | Response Time (secs) | CPU % Busy | # Disk Arms | Disk % Busy |
|---|---|---|---|---|---|---|---|
| **Domino 5.06a V5R1** | | | | | | | |
| 840-2461 24w V5R1 ELAN Mirrored | 100,500 | 27 (2 hubs) | 128GB | 0.07 | 99.3 | 270 | 13.4 |
| 820-2458 4w V5R1 ELAN RAID5 | 12,000 | 4 | 12GB | 0.08 | 71.1 | 43 | 17.8 |
| 820-2458 4w V5R1 TRLAN RAID5 | 6,000 | 1 | 12GB | 0.03 | 38.0 | 20, 2 ASP1,ASP3 | 7.7 , 0.0 ASP1, ASP3 |
| 820-2458 4w V5R1 TRLAN RAID5 , with Transactional Logging Active | 6,000 | 1 | 12GB | 0.04 | 42.3 | 20 , 2 ASP1,ASP3 | 11.4 , 8.9 ASP1,ASP3 |
| 820-2457 2w V5R1 TRLAN RAID5 | 6,750 | 3 | 8GB | 0.07 | 71.0 | 43 | 6.7 |
| 270-2434 2w V5R1 TRLAN RAID5 | 6,750 | 3 | 8GB | NA | 69.6 | 22 | 11.3 |
| 820-2456 1w V5R1 TRLAN RAID5 | 3,250 | 1 | 4GB | 0.11 | 73.1 | 43 | 1.8 |
| **Domino 5.03 V4R5** | | | | | | | |
| 840-2420 24w V4R5 ELAN Mirrored | 75,000 | 27 (2 hubs) | 64GB | 0.28 | 90.9 | 270 | 20.6 |
| 820-2427 4w V4R5 TRLAN RAID-5 | 10,000 | 4 | 12GB | 0.08 | 70.8 | 45 | 11.7 |
| 270-2424 2w V4R5 TRLAN RAID-0 | 6,000 | 2 | 8GB | 0.09 | 84.8 | 12 | 18 |
| 270-2423 V4R5 TRLAN RAID-0 | 3,250 | 1 | 4GB | 0.19 | 87.8 | 6 | 27 |
| 270-2422 V4R5 TRLAN RAID-5 | 1,800 | 1 | 4GB | 0.16 | 82.3 | 11 | 2 |

**Note**:

- Data shown above should not be compared to audited NotesBench results.
- Results may differ significantly from those listed here.
- These measurements are not meant to be interpreted as maximum user data points.
- For the 820-2458 4-way measurement with Transactional Logging active, the logs were created in ASP3.

## 11.12 Web Mail Test Data

The following tables provide a summary of the measured performance data. 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.

*Table 11.4. Web Mail Performance Data*

| Web Mail Serving With Domino on iSeries 400 Server Models | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Model** | **# Active Web Mail Users** | **# Domino Partitions** | **Main Storage** | **Response Time (secs)** | **CPU % Busy** | **# Disk Arms** | **Disk % Busy** |
| **Domino 5.03 V4R5** | | | | | | | |
| 820-2427 4w V4R5 TRLAN RAID-5 | 500 | 1 | 12GB | 0.25 | 13.7 | 45 | 0.1 |
| 270-2424 2w V4R5 TRLAN RAID-0 | 1,000 | 1 | 8GB | 0.26 | 58.8 | 12 | 1 |
| **Domino 5.01 V4R4** | | | | | | | |
| 170-2388 2w V4R4 TRLAN RAID-5 | 500 | 1 | 3.5GB | 0.4 | 54.7 | 10 | 1.4 |

**Note**:
- Data shown above should not be compared to audited NotesBench results.
- Results may differ significantly from those listed here.
- These measurements are not meant to be interpreted as maximum user data points.

# Chapter 12. MQ Series for iSeries

## Introduction

The MQ Series for iSeries product 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, iSeries applications can communicate with other iSeries applications through MQ Series for iSeries, or they can communicate with applications on other platforms by using MQ Series for iSeries and the appropriate MQ Series product(s) for the other platform (HP-UX, OS/390, etc.).

MQ Series supports all important communications protocols, and shields applications from having to deal with the mechanics of the underlying communications being used. In addition, MQ Series 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.

This chapter will discuss performance testing that has been done for Version 5.2 of MQ Series for iSeries and how you can access the available performance data and reports generated from these tests. A brief list of conclusions and results are provided here, although it is recommended to obtain the reports provided for a more comprehensive look at MQ Series for iSeries performance.

## Test Description and Results

Version 5.2 of MQ Series for iSeries includes several performance enhancements designed to significantly improve queue manager throughput and application response time, as well as improve the overall throughput capacity of MQ Series. Measurements were done in the IBM Rochester laboratory with assistance from IBM Hursley to help show the impact of these enhancements, and in some instances, show how Version 5.2 compares to Version 4.2.1.

The workload used for these tests is the standard CSIM workload provided by Hursley to measure performance for all MQ Series platforms. Measurements were done using both client-server and distributed queuing processing. Results of these tests, along with test descriptions, conclusions, recommendations and tips and techniques are available in support pacs at the following URL:
http://www-4.ibm.com/software/ts/mqseries/txppacs/

From this page, you can select to view all support pacs by platform, then choose the AS/400 option to obtain a listing of the documents available for the iSeries. The most current support pac document at this URL is the “MQ Series for iSeries V5.2 - Performance Highlights”. This document contains performance highlights for V5.2 of this product, and includes measurement data, performance recommendations, and performance tips and techniques.

A second performance document will be available as a support pac at this URL by early 4Q01. This document will include performance information for a more comprehensive set of client-server tests, and will also include capacity related information for V5.2 of MQ Series for iSeries.

## Conclusions, Recommendations and Tips

Following are some basic performance conclusions, recommendations and tips/techniques to consider for MQ Series on iSeries. More information is available in the previously mentioned support pacs.

- In general, MQ 5.2 shows a noticeable improvement in peak throughput over MQ 4.2.1 for persistent messaging, both in client-server and distributed messaging environments.

- For applications using non-persistent messaging, the availability of multi-threaded servers in MQ Series 5.2 allows significantly more capacity in terms of peak throughput than was available in MQ 4.2.1.

- Use of a trusted listener process generally results in a reduction in CPU utilization of 5-10% versus using the standard default listener. However, there are other considerations to take into account prior to using a trusted listener. Refer to the "Other Sources of Information" section below to find other references on this subject.

- MQ performance can be sensitive to the amount of memory that is available for use by this product. If you are seeing a significant amount of faulting and paging occurring in the memory pools where applications using MQ Series are running, you may need to consider adding memory to these pools to help performance. More detailed performance information on this will be available in the 4Q01 support pac mentioned above.

- Nonpersistent messages use significantly less CPU and IO resource than persistent messages do because persistent messages use native journaling support on the iSeries to ensure that messages are recoverable. Because of this, persistent messages should not be used where nonpersistent messages will be sufficient.

- If persistent messages are needed, the user can manually create the journal receiver used by MQ Series on a user ASP in order to ensure best overall performance (MQ defaults to creating the receiver on the system ASP). In addition, the disk arms and IOPs in the user ASP should have good response times to ensure that you achieve maximum capacities for your applications that use persistent messages.

## 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 MQ Series:

http://www.ibm.com/software/ts/mqseries/

# Chapter 13. iSeries Linux Performance

## 13.1 Introduction and Executive Summary

iSeries Linux on LPAR (iSeries Linux) is designed to expand the iSeries platform solutions portfolio by allowing customers and software vendors to port existing Linux applications to the iSeries with minimal effort.

### Key Features

- A full-fledged Linux environment. A real Linux operating system in its own, independent, iSeries logical partition (LPAR).
- Similar to pSeries Linux. Whatever runs in pSeries will ordinarily run unchanged, in binary form, on iSeries.
- Normal non-Intel Linux portability considerations.
- Availability of Open Source from source code. If it recompiles and runs on pSeries, it will ordinarily recompile and run on iSeries.
- X-Windows and other Linux/UNIX functionality.
- Intended for environments wanting both OS/400 and Linux functionality.
- Since MacIntosh shares the PowerPC chip with pSeries, some applications known for running on a MacIntosh may run on either pSeries or iSeries with very little work.

This product is not intended to run Linux in a standalone manner. If “just Linux” is wanted for a new machine, there is usually a closely corresponding pSeries hardware solution to the proposed iSeries box. Here, the intent and presumption is that the customer wants and needs function available both on Linux and on OS/400 on a single iSeries hardware package.

### Key Ideas

- Linux on iSeries provides a mechanism to rapidly deploy qualifying Linux and general Open Source applications on iSeries.
- Linux on iSeries provides a mechanism to port certain types of UNIX applications to iSeries.
- Linux on iSeries should also support typical commercial and other non-Open Source solutions, especially if they already run on pSeries Linux.
- Linux on iSeries particularly permits Linux-based middleware to exploit OS/400 function and data in a single hardware package.
- Linux is available only on the more recent iSeries hardware
- Linux is not intended for traditional work (e.g. RPG applications) nor function requiring tight integration with OS/400 facilities.

### Contents

This chapter has the following sections:

1. Basic Model Information. What iSeries models can run Linux? What are the key configuration variables and variations?
2. Basic Configuration and Performance Questions. If one configures a Linux on iSeries partition, what is the effect on the remaining horsepower in the OS/400 partition(s)?
3. Programming environment considerations. When to use Linux, when to use other iSeries facilities to deploy an application. Also includes some technical detail about the Linux architecture on iSeries.
4. General Performance Information

---

## 13.2 Basic Model Information

For various technical reasons, Linux cannot be deployed on all existing iSeries and the older AS/400 machines.

To run Linux at all, a system must:

- **Support logical partitioning (LPAR).** Linux is not part of OS/400. It needs to have its own partition of the system resources, segregated from OS/400 and, for that matter, other Linux partitions if present. This is what a logical partition is for. A special software feature called the Hypervisor keeps each partition operating separately.
- **Support "Guest" Operating Systems** (non-OS/400 operating systems). Part of the iSeries Linux freedom story is to run Linux as Linux, including code from third parties running with root authority and other privilege modes. By definition, such code is not provided by IBM. Therefore, to keep OS/400 and Linux segregated from each other, a few key hardware facilities are needed that are not present on earlier models. (When all partitions run OS/400, the hypervisor's task is simplified, permitting older iSeries and AS/400 to run LPAR).

There is a further division. Some models and processor feature codes can run Linux more flexibly than others. The two key features that not all processors support for Linux are:

- **Shared processors.** This variation of LPAR allows the Hypervisor to share processors between partitions. The goal is that a fixed fraction of a given processor will be owned by some logical partition and another fractional part or parts by another. Thus, a uni-processor might be divided in various fractions between (say) three LPAR partitions. A four way SMP might give 3.9 CPUs to one partition and 0.1 CPUs to another. Not all models and feature codes otherwise capable of running Linux have the additional ability to do shared processors with Linux.
- **Hardware Multi-tasking**. This is controlled by the system value QPRCMLTTSK. Recent AS/400 and iSeries machines have a feature called hardware multi-tasking. With hardware multi-tasking, each CPU has the usual arithmetic processor and the ability to fetch to and from main store. But, with hardware multi-tasking, two sets of registers are present. This allows the operating system to load two different jobs or threads into the same physical CPU, each into one of the independent register sets. When the job represented by one set of registers stalls (typically, because of a cache miss), the other set of registers will take over and executes whatever it can while the first operation completes. This hardware-controlled "back and forth" allows for greater throughput for the majority of applications. Because not all applications profit from hardware multi-tasking, the system has the capacity to turn this facility on and off. Critically, some model and feature codes cannot run a guest operating system, such as Linux, with the hardware multi-tasking feature enabled. Unfortunately, the setting of this value affects nooks and crannies all over the CPU. Accordingly, it requires a reIPL of the primary

partition (and, hence, all partitions) for a change to QPRCMLTTSK to take effect. The probable real-world outcome of this means that if a particular feature code cannot run with QPRCMLTTSK enabled (set to '1'), it will probably run at disabled ('0') all the time, even if Linux is a single, small partition. In typical cases, QPRCMLTTSK set to one improves throughput by 10 to 25 percent, although there are exceptions where it does not profit a given application set at all.

**Processor Tables**

These tables show all the processors capable of running Linux as of V5R1 and which of the above considerations apply. In addition, for those running Linux, some basic performance information is included.

### Model 270 Processors Introduced with V5R1

| 270 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor? | QPRCMLTTSK must be zero? | Num CPUs | MHz | CPW / MCU | CIW |
|---|---|---|---|---|---|---|---|---|
| 2431 | Yes | Yes | Yes | No | 1 | 540 | 465 | 185 |
| 2432 | Yes | Yes | Yes | No | 1 | 540 | 1,070 | 380 |
| 2434 | Yes | Yes | Yes | No | 2 | 600 | 2,350 | 840 |
| 2452 (DSD) | Yes | Yes | Yes | No | 1 | 540 | 3,070 M | 380 |
| 2454 (DSD) | Yes | Yes | Yes | No | 2 | 600 | 6,660 M | 840 |

### Model 270 Processors Introduced with V4R5

| 270 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor? | QPRCMLTTSK must be zero? | Num CPUs | MHz |
|---|---|---|---|---|---|---|
| 2248 | No | No | No | n/a | 1 | 400 |
| 2250 | No | No | No | n/a | 1 | 400 |
| 2252 | No | No | No | n/a | 1 | 450 |
| 2253 | No | No | No | n/a | 2 | 450 |
| 2422 (DSD) | No | No | No | n/a | 1 | 400 |
| 2423 (DSD) | No | No | No | n/a | 1 | 450 |
| 2424 (DSD) | No | No | No | n/a | 2 | 450 |

### Model 820 Processors Introduced with V5R1

| 820 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor ? | QPRCMLTTSK must be zero? | Num CPUs | MHz | CPW / MCU | CIW |
|---|---|---|---|---|---|---|---|---|
| 0150 | Yes | Yes | Yes | No | 1 | 600 | 1,100 | 385 |
| 0151 | Yes | Yes | Yes | No | 2 | 600 | 2,350 | 840 |
| 0152 | Yes | Yes | Yes | No | 4 | 600 | 3,700 | 1,670 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 2435 | Yes | Yes | Yes | No | 1 | 600 | 600 | 200 |
| 2436 | Yes | Yes | Yes | No | 1 | 600 | 1,100 | 385 |
| 2437 | Yes | Yes | Yes | No | 2 | 600 | 2,350 | 840 |
| 2438 | Yes | Yes | Yes | No | 4 | 600 | 3,700 | 1,670 |
| 2456 (DSD) | Yes | Yes | Yes | No | 1 | 600 | 3,110 M | 385 |
| 2457 (DSD) | Yes | Yes | Yes | No | 2 | 600 | 6,660 M | 840 |
| 2458 (DSD) | Yes | Yes | Yes | No | 4 | 600 | 11,810 M | 1,670 |

### Model 820 Processors Introduced With V4R5

| 820 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor? | QPRCMLTTSK must be zero? | Num CPUs | MHz | CPW/ MCU | CIW |
|---|---|---|---|---|---|---|---|---|
| 2395 | Yes | No | No | n/a | 1 | 400 | | |
| 2396 | Yes | No | No | n/a | 1 | 450 | | |
| 2397 | Yes | Yes | No | Yes | 2 | 500 | 2,000 | 680 |
| 2398 | Yes | Yes | No | Yes | 4 | 500 | 3,200 | 1,320 |
| 2425 (DSD) | Yes | No | No | n/a | 1 | 450 | | |
| 2426 (DSD) | Yes | Yes | No | Yes | 2 | 500 | 5,610 M | 680 |
| 2427 (DSD) | Yes | Yes | No | Yes | 4 | 500 | 9,890 M | 1,320 |

### Model 830 Processors

| 830 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor? | QPRCMLTTSK must be zero? | Num CPUs | MHz | CPW | CIW |
|---|---|---|---|---|---|---|---|---|
| 2400 | Yes | Yes | No | Yes | 2 | 400 | 1,850 | 580 |
| 2402 | Yes | Yes | No | Yes | 4 | 540 | 4,200 | 1,630 |
| 2403 | Yes | Yes | No | Yes | 8 | 540 | 7,350 | 3,220 |

### Model 840 Processors Introduced With V5R1 (including Capacity Upgrade On Demand)

| 840 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor ? | QPRCMLTTSK must be zero? | Min CPUs | Max CPUs | MHz | CPW | CIW |
|---|---|---|---|---|---|---|---|---|---|
| 2461 | Yes | Yes | Yes | No | n/a | 24 | 600 | 20,200 | 10,950 |
| 2352 | Yes | Yes | Yes | No | 8 | 12 | 600 | 12,000 | 5,700 |
| 2353 | Yes | Yes | Yes | No | 12 | 18 | 600 | 16,500 | 8,380 |
| 2354 | Yes | Yes | Yes | No | 18 | 24 | 600 | 20,200 | 10,950 |

### Model 840 Processors Introduced With V4R5 (including Capacity Upgrade On Demand)

| 840 Feature | Supports LPAR? | Supports Linux? | Supports Linux Shared Processor ? | QPRCMLTTSK must be zero? | Min CPUs | Max CPUs | MHz | CPW | CIW |
|---|---|---|---|---|---|---|---|---|---|
| 2418 | Yes | Yes | No | Yes | n/a | 12 | 500 | 10,000 | 6,750 |
| 2420 | Yes | Yes | No | Yes | n/a | 24 | 500 | 16,500 | 8,820 |
| 2416 | Yes | Yes | No | Yes | 8 | 12 | 500 | 10,000 | 4,590 |
| 2417 | Yes | Yes | No | Yes | 8 | 12 | 500 | 13,200 | 6,750 |
| 2419 | Yes | Yes | No | Yes | 18 | 24 | 500 | 16,500 | 8,820 |

**Legend (common to all figures, see also Appendicies for more explanations of performance measures):**

CPW -- Maximum Batch CPW rating for regular machines
Maximum Batch CPW for largest number of CPUs for Capacity Upgrade on Demand machines

MCU -- Mail/Calendar users for Domino (DSD) machines. The letter "M" appears after the number when MCU is used as it shares a column with CPW.

CIW -- Compute Intensive Workload. A new measure of overall processing power for CPU intensive workloads. Some Linux work will be better modeled by this than CPW. For Capacity Upgrade on Demand machines, this value is for the maximum number of CPUs.

Min CPUs/Max CPUs. For Capacity Upgrade on Demand Feature codes, the minimum and maximum CPUs available.

Supports .... See previous text.

QPRCMLTTSK must be zero? See previous text.

*Note:* Processors that do not support Linux have their CPW and CIW ratings omitted to bring greater emphasis to the feature codes that support Linux.

---

## 13.3 Basic Configuration and Performance Questions

Since, by definition, the machines are running at least two independent partitions, questions of configuration and performance get surprisingly complicated and quickly.

To keep things simple, consider the following environments:

- A machine with a Linux and an OS/400 partition, both running CPU-bound work with little I/O.
- A machine with a Linux and an OS/400 partition, both running work with much I/O.

The first machine will tend to run as expected. If the OS/400 partition has three of four CPUs and the Linux partition one of four, then the OS/400 "side" will get 3/4 of the CPU and the Linux side 1/4. Since both are CPU-bound, the CIW rating will often give a better guide to execution performance than CPW.

The second machine may be less predictable. This is true for regular applications as well, but it could be much more visible here.

Special problems for I/O bound applications:

- The Linux environment is independent and separate.
- Yet, especially if Linux uses Virtual Disk extensively, it can consume I/O resources on the OS/400 side. In particular, the two partitions may fight each other for disk access. Again, this is normal if one simply were deploying two traditional applications on an iSeries, but the partitioning may make this more noticeable. Initially, in fact, one may not be able to attribute the I/O to "anything" running on the OS/400 side, since the various OS/400 performance tools don't know about any other partition, much less a Linux one.
- Some of the problems with Virtual Disk can also occur with an Integrated xSeries Server, since it (like Linux) can use a Virtual Disk.

Some solutions:

- In many cases, awareness of this situation may be enough. After all, new applications are deployed in a traditional OS/400 environment all the time. These often fight existing, concurrent applications for the disk.
- Existing guidelines suggest that disk utilization be kept below 42 per cent for non-load source units. If this is done, the overall installation will be within guidelines and overall performance should ordinarily be acceptable.
- However, since Linux is in its own partition, and doesn't support OS/400 notions of subsystem and job control, awareness may not be enough. Alternate solutions include native disk and, possibly, segregating the Linux Virtual Disk (using OS/400 Network Storage objects) into a separate ASP.

---

## 13.4 Programming Environment Considerations

Since it is obviously true that this environment cannot run Intel-based binaries, some Linux applications will need recompiling for PowerPC. Open source, by definition, will have the source available.

Therefore, recreating open source applications on the iSeries Linux partition directly from the source code distribution is also possible, provided the ordinary dependency management is satisfied (e.g. compatible kernel levels) and also that the application in question understands and accounts for the issue of byte order (big versus little endian). iSeries Linux, being a true Linux operating system, will deal with the normal issues of binary formats for applications, including the output of the PowerPC version of the gcc compiler, linker, and other compatible tools.

Note carefully that Linux applications running on pSeries or zSeries are already available in the same byte order as Linux on iSeries. It is only applications sourced from xSeries or other 'little endian' machines which may have byte order problems. In fact, makefile settings for the pSeries should ordinarily be satisfactory for source recompilation on iSeries. Code developed for or at least running on an Apple MacIntosh (but running Linux) should also port with little difficulty.

See "Characteristics of Application Candidates for iSeries Linux" for more information.

### iSeries Linux Technical Overview

iSeries Linux is a program-execution environment on the iSeries system that provides a traditional memory model (not single-level store) and allows direct access to machine instructions (without the mapping of MI

architecture). Because they run in their own partition on a Linux Operating System, programs running in iSeries Linux do have direct access to the full capabilities of the user-state and even most supervisor state architecture of PowerPC. They do *not* have access to the single level store and OS/400 facilities. To reach OS/400, its single level store, and its functions, requires some sort of machine-to-machine interface, such as sockets. A high speed virtual LAN is available to expedite and simplify this communication.

Storage for Linux comes from two sources: Native and Virtual disks (the latter implemented as OS/400 Network storage). Native access is provided by allocating ordinary iSeries hard disk to the Linux partition. Linux can, by a suitable and reasonably conventional mount point strategy, intermix both native and virtual disks. The virtual disk is analogous to some of the less common Linux on Intel distributions where a Linux file system is emulated out of a large DOS/Windows file, except that on OS/400, the storage is automatically "striped" to multiple disks.

Linux partitions can also have virtual or native local area networks. Typically, a native LAN would be used for communications to the outside world (including the next fire wall) and the virtual LAN would be used to communicate with OS/400. In a full-blown DMZ ("demilitarized zone") solution, one Linux partition could provide a LAN interface to the outer fire wall, could then talk to a second providing the inner fire wall, and then the second Linux partition could use virtual LAN to talk to OS/400.

iSeries Linux currently provides support for 32-bit PowerPC Linux applications.

**iSeries Linux Run-time Support**

Linux brings significant support including X-Windows and a large number of shells and utilities. Languages other than C (e.g. Perl, Python, etc.) are also supported.

Applications running in iSeries Linux work in ASCII. Neither the gcc compiler nor any other Linux-based code generator support EBCDIC. When talking from Linux to OS/400, care must be taken to deal with ASCII/EBCDIC questions. However, for a great fraction of the ordinary Internet and other sockets protocols, it is the OS/400 that is required to shoulder the burden of translation -- the Linux code can and should supply the same ASCII information it would provide any other system in a given protocol.

iSeries Linux, being real Linux, has as much support for Unicode as the application itself provides. Generally, the Linux kernel itself currently has no support for Unicode. This can complicate the question of file names, for instance, but no more or no less than any other Linux environment.

**iSeries Linux Development Environment**

iSeries Linux development could be performed natively under iSeries. However, there is no direct support available from IBM. Tools such as KDevelop and others, being themselves ordinary X-Windows functions, might be made available by a vendor or by the open source community generally. Whatever is available for pSeries should generally work on iSeries. The main issue would be compatible levels of kernels and the usual application dependencies, whatever they may be.

It should also be possible to do the majority of development on another Linux platform and send the final source tree to the iSeries for final compilation and test. A pSeries would be a great choice for this. If Linux on Intel is chosen for this scheme, special attention should be paid to byte order in the detailed code

and test. If this is managed well, little difficulty will be seen when the final compile-and-test is performed on iSeries.

**Characteristics of Application Candidates for iSeries Linux**

Most candidates for an iSeries for Linux port will already run on Linux. Therefore, they will virtually always want to run on iSeries Linux.

But, what about code form other sources, such as other UNIX or possibly even NT? What about applications with source available both from Linux and one of these other environments?

When planning to port an application to the iSeries there are three choices:

1. The iSeries Integrated Language Environment (ILE),
2. PASE (Portable Applications Solution Environment)
3. iSeries Linux.

Here are some of the reasons to choose iSeries Linux:

1. If the UNIX/AIX APIs your applications uses are already supported by iSeries Linux, then there is very little application porting to be done. Being a complete Linux, a great fraction of the UNIX interfaces are present, though details need to be checked since Linux is not UNIX.
2. iSeries Linux readily allows the use of UNIX type build processes, which is especially useful when you have an existing, complicated build process.
3. iSeries Linux supplies support for *fork* and *exec*, which does not currently exist on OS/400 (except through *spawn* which is significantly different).
4. iSeries Linux is superb at satisfying dependencies on an ASCII character set and for satisfying dependencies on X-Windows support.
5. iSeries Linux fully supports ANSII C, C++ and FORTRAN. It also supports the *de facto* industry 32 bit programming model.
6. Shell programming is supported.

Most of these reasons are shared with PASE.

Reasons to go with Linux over PASE:

1. Dependencies on other Linux tools and middleware.
2. Desire to port the code to other Linux environments.

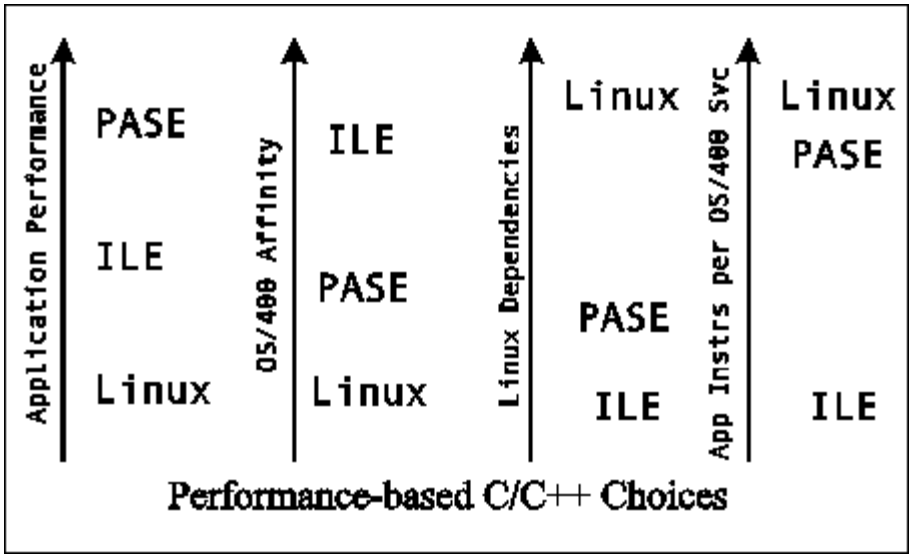
Reasons to go with PASE over Linux:

1. Available on more AS/400 platforms and all iSeries.
2. The AIX compilers usually produce the best optimized code for applications. The difference is especially noticeable in floating point-based applications.
3. Desire to port the code to AIX.

Reasons to go with ILE C/C++ include:

1. Tight integration with OS/400 and all its facilities. The more frequently OS/400 services are accessed, the more important this becomes. Conversely, if OS/400 facilities are accessed infrequently (including protocols that deliver many "records" per trip), the case for Linux is strengthened.
2. Requirements to deal with EBCDIC.
3. Better application performance than Linux, most likely seen in integer-based applications.
4. Available on all RISC OS/400 models.

From a performance perspective, the following chart express the key decision points (arrow is in the direction of "increasing" as in "Application Performance is increasingly important"):




The tradeoffs are:

1. Application performance favors PASE and ILE over Linux. Note, conversely, that for applications whose dominant feature is simply invoking OS/400 (whether directly or indirectly via, say, sockets) the sensitivity to "which application environment to choose" is reduced all around.
2. High affinity to OS/400 services favors ILE over PASE and PASE over Linux. If access to OS/400 facilities is frequent, ILE is obviously the most tightly integrated and has smallest overhead per access. Functionally, this will mean simpler, local access versions more complicated and indirect interfaces such as sockets.
3. Linux dependencies (e.g. Non-C/C++ based tools like "scripts") favors Linux.
4. High computation per access of OS/400 resources favors PASE and Linux. As the portion of computation in the would-be PASE or Linux function grows, affinity to OS/400 matters less and less and the need and even desire to use the ILE environment diminishes. Moreover, Linux has an added advantage here in that any system services which Linux can satisfy presumably are never are redirected to OS/400. Thus, one goes beyond simply considering "application" performance to include Linux middleware and kernel function as well. In PASE, similar services would be emulated by OS/400. Note that 2 and 4 are not entirely mutually exclusive

Functional considerations and programming costs may be more important than performance for this decision. Application providers can obtain much more support for these decisions and for porting their applications to iSeries from PartnerWorld for Developers at http://www.as400.ibm.com/developer/ .

## 13.5 Performance Test Results

A limited number of  performance related tests have been conducted to date, comparing the performance of iSeries Linux to other environments on iSeries and to compare performance to similarly configured (especially CPU MHz) pSeries running the application in an AIX environment.

### Running with OS/400 in another partition

A key question deserving an answer is simply this:  "If I run a Linux partition, what is the effect on the OS/400 part of my workload?"

For instance, suppose a customer, having spare capacity on a 4-way iSeries box, decides to create a Linux partition and give one of the four CPUs up to Linux.  Will that customer get 3/4 of the CPW rating on the OS/400 "side" of the resulting system?

For CPU intensive workloads on both sides, the answer is "yes."  The iSeries hypervisor ensures that one CPU is available to Linux for whatever it consumes, and the OS/400 share the remaining three in the usual manner.  In this case, the normal LPAR expectations are realized.

For disk intensive workloads, the answer is more complicated.  To the extent virtual disk in particular is used, and to the extent virtual LAN is used, the OS/400 will end up spending some resources dealing with the virtual I/O.  It will also simply have to wait sometimes to do its own work with the underlying physical disks in the case of virtual disk on the Linux side (a virtual disk is, after all, real storage on the OS/400 side).  Now, if this was just another new application on the OS/400 side, there would be nothing remarkable about this.  Adding a new application to an existing set could also cause problems if it were disk intensive and the old applications were disk intensive as well.

Because these are independent environments, the normal management tools (e.g. OS/400 subsystems, activity limits, etc.) are not available.  It may be more helpful than usual to either give Linux its own disks ("native" disks) or to put the Linux virtual disk (Network Storage objects) in their own ASP.  This will reduce or eliminate interference between OS/400 and Linux applications.  The advantage of Native disks is that Linux controls them *in toto*.  The advantage of virtual disk in its own ASP is that interference is minimal and you gain a unique, iSeries-based advantage in that network storage automatically functions as a 'striped' disk -- if the ASP has six disks, then every file is divided evenly between the six disks.

### Computational Performance

Tradeoffs between when to choose PASE, Linux, and ILE C/C++ have already been discussed.

Generally, for integer-based applications (general commercial):

- PASE gives the fastest integer performance.
- ILE C/C++ is usually next
- Linux is last.

Ordinarily, all would be within a binary order of magnitude of each other. The difference is close enough that ILE C/C++ sometimes is faster than PASE. Linux usually lags slightly more, but is usually not significantly slower.

Generally, for applications dominated by floating point, the rankings change somewhat.

- PASE almost always gives the fastest performance.
- Linux and ILE C/C++ often trail substantially. In one measurement, Linux was 2.4 times slower than PASE.

ILE C/C++ performance will be closer to Linux than to PASE. Note carefully that most commercial applications do not feature floating point.

### Gcc and Optimization (gcc -O3)

The gcc compiler (and other seemingly different compilers that invoke gcc's code generation, such as g++) is used for a great fraction of Linux applications and the Linux kernel.

Generally speaking, RISC architectures have assumed that the final, production version of an application would be deployed at a high optimization. Therefore, it is important to specify the best optimization level (gcc option -O3) when compiling with gcc or any gcc derivatives. When debugging (-g), optimization is less important and even counterproductive, but for final distribution, optimization often has dramatic performance differences from the base case where optimization isn't specified.

Programs can run twice as fast or even faster at high versus low optimization. It may be worthwhile to check and adjust Makefiles for performance critical, open source products. Likewise, if compilers other than gcc are deployed, they should be examined to see if their best optimizations are used.

One should check the *man* page (*man gcc* at a command line) to see if other optimizations are warranted. Some potentially useful optimizations are not automatically turned on because not all applications may safely use them.

#### Follow-on Performance Information

Linux, and important products running upon it, can deployed independently of OS/400 and its releases. Further performance information may become available after this document is published as this document is synchronized with the OS/400 release schedule.

For the latest performance information see: http://www.ibm.com/eserver/iseries/linux/performance.html

# Chapter 14. DASD Performance

This chapter discusses DASD performance at the DASD device level, DASD subsystem level and the AS/400 / iSeries system level. Performance of various types of DASD configurations is characterized for both batch and interactive applications. Performance comparisons between different types of DASD protection schemes (such as mirroring and RAID) are presented along with the performance of DASD configurations without protection. DASD compression performance is characterized and compared to uncompressed performance. The performance benefits from system level Expert Cache and the DASD subsystem level Extended Adaptive Cache are also discussed.

## 14.1 Device Performance Characteristics

This section compares the performance of the Internal DASD Subsystems based on the 65x2 RAID Controllers or 6530 Storage Controller with the external 9337 Disk Array Subsystem using a system configured with an equivalent amount of DASD capacity. This section also contains performance characteristics for the 6532, 6533, 2726, 2740, 2741, 2748 (4748), 2763 and new 2778 (4778) RAID Controllers and 6751, 6754 and 9728 MFIOPs. The performance is based on measurements and modeling done in the development laboratory. Because the performance of the AS/400 system is dependent on many factors, these characteristics are very general in nature. To assess the various configuration options, one of the capacity planning tools should be used.

The performance characteristics of Internal DASD is listed in Table 14.1 and Table 14.2 below and the performance characteristics of External DASD is listed in Table 14.3. The tables do not list all of the feature codes, but it does provide performance information for most of the disk configurations. For example, the 6522 IOP has the same performance characteristics as the 6502 IOP. For a description of the DASD models supported by the 6502,6512, 6530, 6532, 6533, 6751, 6754, 2726, 2740, 2741, 2748, 2778, 2763 and 9728 IOP/IOAs refer to Appendix C, “DASD IOP/IOA Device Characteristics”.

In the tables, the following measures of performance are listed.

**Service Time** is the time required to perform the "Interactive op" described in the next paragraph. The time starts with the request from the CPU to the Disk IOP and the time stops when the data is in main storage (read) or when the data is on the disk or in the write cache (write). Queuing time is not included.

**Interactive Ops/Sec** is an estimate of the number of IOs that can be done at 40% utilization using the service time calculated for the previous column. If the disk model contains 2 arms, this number only reflects the capacity of one arm. At 40%-50% utilization, the disk arms are at the "knee of the curve". As utilization exceeds the "knee of the curve", response time increases significantly and becomes erratic. We assume the following:

- 40% arm utilization
- 7KB transfer size
- 70% read and 30% write
- 80% 1/3 seek and 20% 0 seek

**Interactive Rel** is the Relative Interactive performance of the disk drives. This column is the same as the INTERACTIVE Ops/Sec column except that the numbers are normalized to 1.0.

**Batch Hours** is an estimate of how long batch type applications would execute. The duration of many batch type jobs depends on the performance of the disk. For ease of understanding, the numbers are normalized to 8 hours assuming the slowest disk drive is used. We assume the following:

- 75% of the batch job time is disk IO
- Average of 4KB, 8KB and 16KB transfer sizes
- 70% read and 30% write
- 20% 1/3 seek and 80% 0 seek

**Ops/Sec/GB** is an estimate of how many system physical disk IOs per second per usable GB of space that the specific model of DASD can perform when the arm is 40% utilized. The write cache effectiveness reduces the volume of writes that the physical disk drive must support. For the 9337-2xx models, the write cache effectiveness is assumed to be 45% and for the 9337-4xx and 9337-5xx models it is assumed to be 65%. For the 6502 IOP, the write cache effectiveness is assumed to be 55%. For the 6512, 6532, 6533, 6751, 6754, 2726, 2740 and 2741 IOP/IOAs, the write cache effectiveness is assumed to be 65%. The write cache effectiveness is assumed to be 67% for the 2763 IOA and 70% for the 2748 IOA and 75% for the new 2778 IOA. We use the service time required to physically write the record to DASD. The service time contained in column four included the faster write completion that resulted when the write was safely in the write cache.

*Table 14.1. DASD Performance - Internal DASD*

| Disk | | Number | Service | Interactive | | Batch | Ops/Sec/GB | |
|---|---|---|---|---|---|---|---|---|
| Model | MB | of Arms | Time | Ops/Sec | Rel | Hours | Base | RAID |
| 6502-6605 | 1031 | 1 | 8.7 | 46.0 | 2.3 | 3.6* | 38 | 29 |
| 6502-6606 | 1967 | 1 | 8.8 | 45.5 | 2.3 | 3.6* | 20 | 15 |
| 6502-6607 | 4194 | 1 | 8.8 | 45.5 | 2.3 | 3.6* | 9 | 7 |
| 6502-6713 | 8589 | 1 | 9.1 | 44.0 | 2.2 | 3.6* | 4 | 3 |
| 6502-6714 | 17548 | 1 | 9.1 | 44.0 | 2.2 | 3.6* | 2 | 2 |
| 6512-6605 | 1031 | 1 | 8.2 | 48.8 | 2.4 | 3.4* | 43 | 36 |
| 6512-6606 | 1967 | 1 | 8.3 | 48.2 | 2.4 | 3.4* | 22 | 18 |
| 6512-6607 | 4194 | 1 | 8.3 | 48.2 | 2.4 | 3.4* | 10 | 8 |
| 6512-6713 | 8589 | 1 | 8.6 | 46.5 | 2.3 | 3.4* | 5 | 4 |
| 6512-6714 | 17548 | 1 | 8.6 | 46.5 | 2.3 | 3.4* | 2 | 2 |
| 6530-6605 | 1031 | 1 | 11.4 | 35.1 | 1.7 | 3.9 | 34 | |
| 6530-6606 | 1967 | 1 | 11.6 | 34.5 | 1.7 | 3.9 | 18 | |
| 6530-6607 | 4194 | 1 | 11.6 | 34.5 | 1.7 | 3.9 | 8 | |
| 6530-6713 | 8589 | 1 | 12.0 | 33.3 | 1.7 | 4.0 | 4 | |
| 6530-6714 | 17548 | 1 | 12.0 | 33.3 | 1.7 | 4.0 | 2 | |

**Note**:

The 6502 and 6512 IOP write cache is only used for 1 GB and larger DASD. The write cache is NOT used for any 400 MB or 988 MB DASD that are attached.

* For the 6502 and 6512 IOPs in RAID mode, most batch jobs will run nearly as fast as if they were run in 'base' or mirrored mode. Only in extreme cases will the RAID mode cause degradation. An example is when there are sequences of hundreds of writes to a single IOP in a short period of time.

*Table 14.2. DASD Performance - Internal DASD (continued)*

| Disk Model | MB | Number of Arms | Service Time | Interactive | | Batch Hours | Ops/Sec/GB | |
|---|---|---|---|---|---|---|---|---|
| | | | | Ops/Sec | Rel | | Base | RAID |
| 6533-6605 | 1031 | 1 | 8.0 | 50.0 | 2.5 | 3.4* | 43 | 36 |
| 6533-6606 | 1967 | 1 | 8.1 | 49.4 | 2.5 | 3.4* | 23 | 19 |
| 6533-6607 | 4194 | 1 | 8.1 | 49.4 | 2.5 | 3.4* | 11 | 9 |
| 6533-6713 | 8589 | 1 | 8.4 | 47.6 | 2.4 | 3.4* | 5 | 4 |
| 6533-6717 | 8589 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 6 | 5 |
| 6533-6714 | 17548 | 1 | 8.4 | 47.6 | 2.4 | 3.4* | 2 | 2 |
| 6533-6718 | 17548 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 3 | 2 |
| 2748-6607 | 4194 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 12 | 10 |
| 2748-6713 | 8589 | 1 | 7.5 | 53.3 | 2.7 | 3.2* | 6 | 5 |
| 2748-6717 | 8589 | 1 | 6.5 | 61.5 | 3.1 | 3.0* | 7 | 5 |
| 2748-6714 | 17548 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 3 | 2 |
| 2748-6718 | 17548 | 1 | 6.5 | 61.5 | 3.1 | 3.0* | 3 | 3 |
| 2778-6607 | 4194 | 1 | 7.0 | 57.1 | 2.8 | 3.1* | 13 | 10 |
| 2778-6713 | 8589 | 1 | 7.3 | 54.8 | 2.7 | 3.1* | 6 | 5 |
| 2778-6717 | 8589 | 1 | 6.3 | 63.5 | 3.2 | 3.0* | 7 | 5 |
| 2778-6714 | 17548 | 1 | 7.0 | 57.1 | 2.8 | 3.1* | 3 | 2 |
| 2778-6718 | 17548 | 1 | 6.3 | 63.5 | 3.2 | 3.0* | 3 | 3 |
| 2763-6607 | 4194 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 12 | 10 |
| 2763-6713 | 8589 | 1 | 7.5 | 53.3 | 2.7 | 3.2* | 6 | 5 |
| 2763-6717 | 8589 | 1 | 6.5 | 61.5 | 3.1 | 3.0* | 7 | 5 |
| 2763-6714 | 17548 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 3 | 2 |
| 2763-6718 | 17548 | 1 | 6.5 | 61.5 | 3.1 | 3.0* | 3 | 3 |
| 6754-6605 | 1031 | 1 | 8.0 | 50.0 | 2.5 | 3.4* | 43 | 36 |
| 6754-6606 | 1967 | 1 | 8.1 | 49.4 | 2.5 | 3.4* | 23 | 19 |
| 6754-6607 | 4194 | 1 | 8.1 | 49.4 | 2.5 | 3.4* | 11 | 9 |
| 6754-6713 | 8589 | 1 | 8.4 | 47.6 | 2.4 | 3.4* | 5 | 4 |
| 6754-6717 | 8589 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 6 | 5 |
| 6754-6714 | 17548 | 1 | 8.4 | 47.6 | 2.4 | 3.4* | 2 | 2 |
| 6754-6718 | 17548 | 1 | 7.2 | 55.6 | 2.8 | 3.1* | 3 | 2 |
| 9728-6605 | 1031 | 1 | 10.9 | 36.7 | 1.8 | 3.8 | 38 | |
| 9728-6606 | 1967 | 1 | 11.0 | 36.4 | 1.8 | 3.8 | 18 | |
| 9728-6607 | 4194 | 1 | 11.0 | 36.4 | 1.8 | 3.8 | 9 | |
| 9728-6713 | 8589 | 1 | 11.3 | 35.4 | 1.8 | 3.9 | 4 | |
| 9728-6717 | 8589 | 1 | 9.7 | 41.2 | 2.1 | 3.6 | 5 | |
| 9728-6714 | 17548 | 1 | 11.3 | 35.4 | 1.8 | 3.9 | 2 | |
| 9728-6718 | 17548 | 1 | 9.7 | 41.2 | 2.1 | 3.6 | 2 | |

**Note**:

* The 6533 IOP has slightly better performance than the 6532 IOP but will usually be noticeable only at higher throughput ranges. The 2741 IOA has slightly better performance than the 2726 IOA but will usually be noticeable only at higher throughput ranges. The 6754 MFIOP has the same performance relationship with the 6751 MFIOP. The ***2740 IOA*** (which is targeted for smaller systems) has similar performance to the 2726 IOA over typical operating ranges, but has slightly slower performance at higher throughput ranges. The ***2763 IOA*** is an improved upgrade for the 2740 and has performance similar to the 2748. The ***new 2778 IOA*** is an upgrade for the 2748 and has up to 10% performance improvement.

* For the 6532, 6533, 6754, 2726, 2740, 2741, 2748, 2763 and new 2778 IOP/IOAs in RAID mode, most batch jobs will run nearly as fast as if they were run in 'base' or mirrored mode. Only in extreme cases will the RAID mode cause degradation. An example is when there are sequences of hundreds of writes to a single IOP in a short period of time.

These IOP/IOAs are also capable of attaching Ultra-SCSI (40 MB/sec bus) versions of the 6606, 6607, 6713, and 6714 DASDs. These devices can improve performance for workloads characterized by large disk I/O operations. The 2748, 2778 and 2763 IOAs are capable of supporting the SCSI Wide-Ultra2 (80 MB/sec) bus.

*Table 14.3. DASD Performance - External DASD*

| Disk Model | MB | Number of Arms | Service Time | Interactive Ops/Sec | Interactive Rel | Batch Hours | Ops/Sec/GB Base | Ops/Sec/GB HA |
|---|---|---|---|---|---|---|---|---|
| 9337-210 | 1084 | 2 | 12.4 | 32.3 | 1.6 | 4.3* | 50 | 36 |
| 9337-215 | 1084 | 2 | 9.8 | 40.8 | 2.0 | 3.9* | 63 | 46 |
| 9337-220 | 1940 | 2 | 12.5 | 32.0 | 1.6 | 4.3* | 28 | 20 |
| 9337-225 | 1940 | 2 | 11.0 | 36.4 | 1.8 | 4.0* | 32 | 23 |
| 9337-240 | 7868 | 4 | 11.3 | 35.4 | 1.8 | 4.0* | 15 | 11 |
| 9337-420 | 3880 | 4 | 8.6 | 46.5 | 2.3 | 3.5* | 43 | 36 |
| 9337-440 | 7868 | 4 | 8.8 | 45.5 | 2.3 | 3.5* | 21 | 18 |
| 9337-480 | 16776 | 4 | 9.1 | 44.0 | 2.2 | 3.6* | 10 | 8 |
| 9337-540 | 7868 | 4 | 8.6 | 46.5 | 2.3 | 3.5* | 21 | 18 |
| 9337-580 | 16776 | 4 | 8.6 | 46.5 | 2.3 | 3.5* | 10 | 8 |
| 9337-590 | 34356 | 4 | 8.9 | 44.9 | 2.2 | 3.6* | 5 | 4 |

**Note**:

* For the 9337-2xx, 9337-4xx, and 9337-5xx models in HA mode, most batch jobs will run nearly as fast as if they were run in 'base' or mirrored mode. Only in extreme cases will the RAID mode cause degradation. An example is when there are sequences of hundreds of writes to a single IOP in a short period of time.

## Conclusions / Recommendations

The 6532, 6751 and 2726 DASD IOP/IOAs have similar performance characteristics. The 6533 IOP, 2741 IOA and 6754 MFIOP have slightly better performance characteristics, which are more beneficial at higher throughput ranges. The 2740 IOA (which is targeted for smaller systems) has similar performance characteristics to the 2726 IOA over typical operating ranges, but has slightly slower performance at higher throughput ranges. These IOP/IOAs are also capable of attaching Ultra-SCSI (40 MB/sec bus) versions of the 6606, 6607, 6713 and 6714 DASDs. These devices can improve performance for workloads characterized by large disk I/O operations. The 2748 PCI IOA has better performance characteristics, a larger write cache (26MB), and is capable of supporting the SCSI Wide-Ultra2 (80 MB/sec) bus. The 2763 PCI IOA (which is targeted for AS/400 models 270 and 820) has performance characteristics similar to the 2748, a 10MB write cache, and is capable of supporting the SCSI Wide-Ultra2 (80 MB/sec) bus. The new 2778 PCI IOA has performance characteristics up to 10% better than the 2748, a 26MB write cache (data in cache is compressed - effective size can be up to 104MB) , and is capable of supporting the SCSI Wide-Ultra2 (80 MB/sec) bus.

The DASD that are used in the Internal DASD Subsystems have read ahead buffers that can provide performance advantages. Like the 9337, each of these DASD has a 512K buffer. The buffer is allocated into multiple segments that are larger than 32K each. Read ahead data from recent IOs are kept in these buffer segments. Depending on the data access patterns, it is possible that the data needed is already contained in a buffer segment. If so, no physical access to the DASD is required. Depending on your data access patterns, this can significantly improve performance. Our analysis of several specific customer installations indicates that 10% to 30% of their DASD IO for "interactive" transactions would have already been contained in the read ahead buffer. For "batch" type jobs, 25% to 45% of their DASD IO would have already been contained in the read ahead buffer. The RAMP-C workload being used in this section has less than 10% of it's DASD IOs already in the read ahead buffer.

For the 9337-2xx, 9337-4xx, 9337-5xx, 65x2 (also 2726, 2740, 2741, 2748, 2778, 2763, 6533, 6751 and 6754) models in RAID mode, most batch jobs will run nearly as fast as if they were run in "base" mode or

mirrored mode. Only in extreme cases will the RAID mode cause degradation. An example of the extreme case is when there are sequences of hundreds of writes to a single 9337 or 65x2 in a short period of time.

You must ensure that you have enough arms to support the volume of DASD IOs that your customer will require. In some situations, using the larger capacity DASD may result in an insufficient number of arms to handle the required DASD IO volume. The Capacity Planning tools should be used to verify your configuration.

The recommended threshold for maximum DASD utilization for 1 arm configurations is higher than the threshold for multiple arm configurations. The reason for the lower recommendation for multiple arms is that it is assumed that when 2 or more arms have an average utilization of 40%, some of the arms may be at the 50% - 55% range while others will be lower. QSIZE400 and BEST/1 allow a 1 arm configuration to reach 55% before they recommend that an additional DASD be added.

Consider the following example. Assume you are configuring a system and need approximately 4000 MB of DASD space. You have the choice of 4 x 988MB or 2 x 1967MB. The 4 x 988MB configuration will support approximately 70% more DASD IOs as the 2 x 1967MB configuration. Because there is a maximum number of DASD devices that can be attached to each model, using the larger drives will allow more MB of DASD to be configured on your system.

The Performance Monitor (STRPFRMON command) captures additional performance data (buffer hits, etc.) for the attached DASD. For V5R1 and following, Collection Services will be used to collect this performance data. This data is available in the QAPMDISK performance data file and is documented in Appendix A of the *AS/400 Work Management V4R3* (SC41-5306-02).

## 14.2 DASD Performance - Interactive

The implementation of the 4KB page size on RISC will improve system DASD IO efficiency. As a result of the larger page size, some DASD subsystem interactive Ops/Sec/GB ranges will appear lower than on IMPI.

Some DASD system performance charts included for RISC may differ from similar charts published for IMPI. These performance differences can be attributed primarily to the following:

- Differences in system processor power
- Differences in main storage configurations
- Differences in system page size
- Differences in allocation of data and programs on DASD.

Therefore, direct comparisons between RISC and IMPI system DASD performance charts are not recommended.

## DASD Subsystem Performance - Base or Mirrored

The following bar graphs compare the service times for the AS/400 DASD subsystem offerings. The IO operations being performed are 7KB transfer size, 70% are reads and 30% are writes, and 80% require a seek over 1/3 of the disk surface while 20% require no seek. Queuing time is not included.

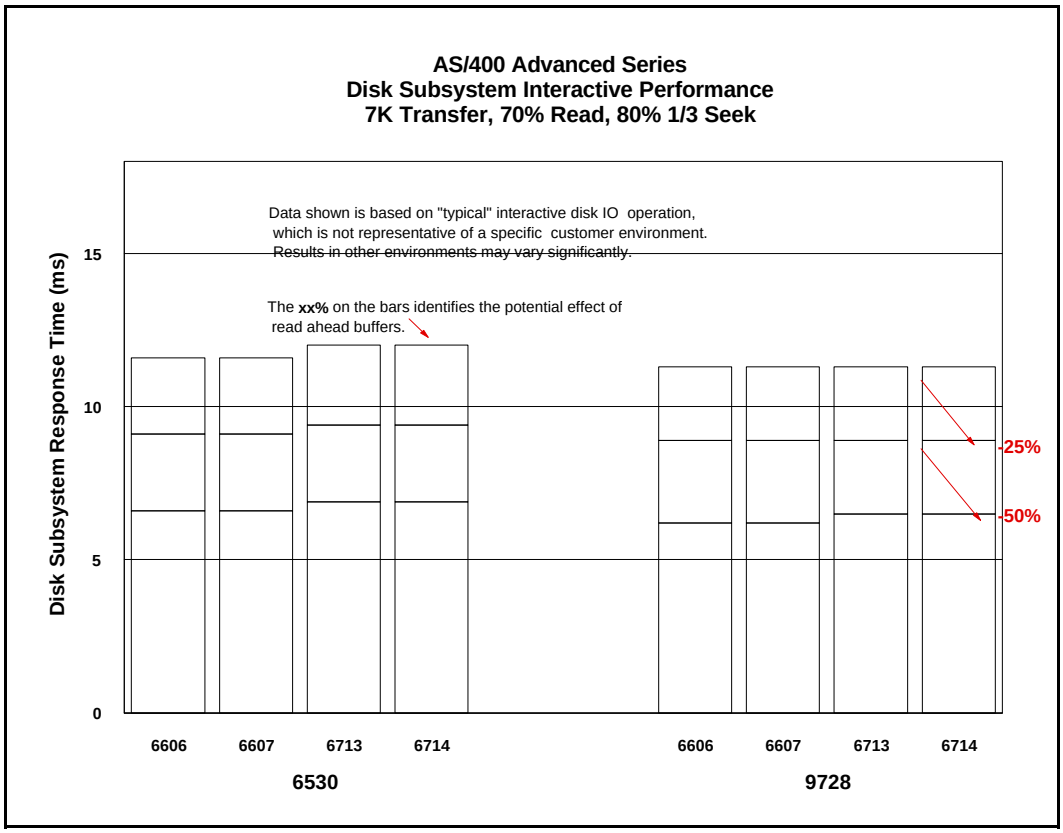



Figure 14.1. DASD Subsystem Performance / Non-Raid Capable - Base Mode

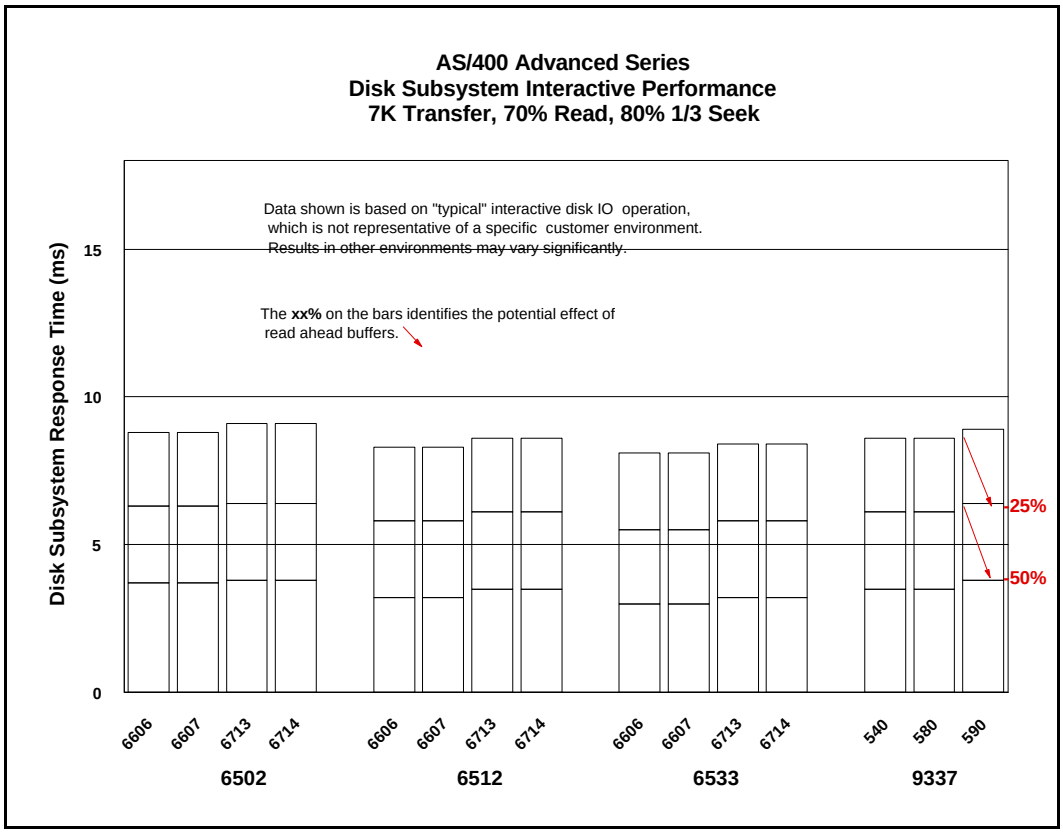



Figure 14.2. DASD Subsystem Performance / Raid Capable - Base Mode

## Conclusions / Recommendations

- The performance of 6606, 6607, 6713 and 6714 disks is faster with the 6533 IOP than with the 6512 IOP. The 6754, 6751, 6532, 2726, 2740 and 2741 DASD IOP/IOAs have performance characteristics similar to the 6533 IOP over typical operating ranges.

- The performance with 6606, 6607, 6713 and 6714 disks is faster than the previous DASD types for all the subsystems.

- The 9728 subsystem performance is faster than the 6530 subsystem for the same type of DASD.

- The 6512 subsystem performance is better than the 6502 subsystem for the same type of DASD. This is due primarily to a faster processor and a larger 4 MB Write cache in the 6512.

- The 6512 subsystem performance is slightly better than the 9337-5xx subsystem for the same type of DASD.

- The 6502 subsystem performance is significantly better (32%) than the 6530 subsystem for the same type of DASD. This is due primarily to the 2 MB Write cache in the 6502.

- "RAID Capable" DASD subsystems are faster in base mode than "Non-RAID Capable" due to their write cache.

- The potential effect of read-ahead buffers are shown for the cases of having 25% and 50% of the total disk operations already in the read ahead buffer. Depending on the data access patterns, the buffers may provide significant performance improvements.

- The above conclusions hold for batch environments also. For actual batch performance results refer to Table 14.1, Table 14.2, and Table 14.3.

## AS/400 System Interactive Performance - Base

The following graph compares the relative interactive performance of an AS/400 model 510/2144 configured with 33.5GB of internal or external DASD. The internal load source drive was ignored for this comparison chart. The curves characterize what may occur on either a 'Base' configuration or a mirrored configuration. The graph compares the 9337-580 model with the 65x2-4GB models for a commercial interactive environment.

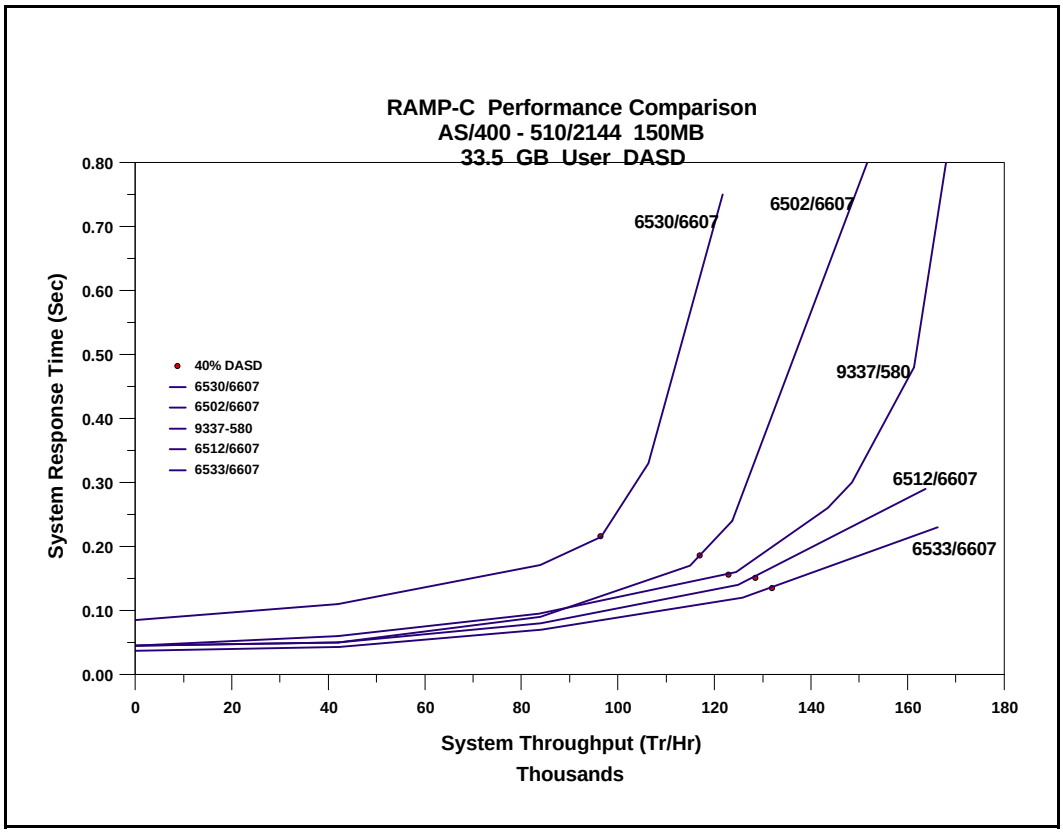



Figure 14.3. System Interactive Performance - Base Mode

## Conclusions / Recommendations

- The 6533/6607 DASD provides better interactive performance than the 6512/6607 DASD. The 6754, 6751, 6532, 2726, 2740 and 2741 DASD IOP/IOAs have performance characteristics similar to the 6533 IOP over typical operating ranges.

- The 6512/6607 DASD provides better interactive performance than the 9337-580 DASD.

- The 6512/6607 DASD provides better interactive performance than the 6502/6607 DASD, especially at higher system throughput.

- Performance with the 6502/6607 DASD is slightly better than the 9337-580 DASD for lower throughput, but becomes worse at higher throughputs.

- The 65x2 and 9337-5xx configurations have reduced volumes of physical disk IO due to the write cache. The write cache also greatly improves the service time for write ops.

- The 65x2 write cache provides a significant performance advantage over the 6530. When a write is requested to a 65x2, the 65x2 writes the data to the write cache and to the nonvolatile cache backup and the application is allowed to continue. Through a combination of the write cache and 65x2 nonvolatile memory, the 65x2 ensures the integrity of the data even if a failure should occur.

- This graph is based on RAMP-C workload. Other environments may vary significantly.

- The RAMP-C benchmark's data access patterns are intentionally random, therefore, the read-ahead buffers provided only minimal benefit for RAMP-C. Depending on your data access patterns, the DASD read ahead buffers may provide significant performance improvements.

- Similar results may occur on other AS/400 models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster AS/400 CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

**AS/400 System Interactive Performance - Mirrored versus Base**

The following graph compares the relative interactive performance of an AS/400 model 510/2144 configured with 8.4 GB of User DASD. The graph compares a mirrored environment with 16 arms to a base (not mirrored) environment with 8 arms. It also shows the system performance effects during the resync of a single arm.

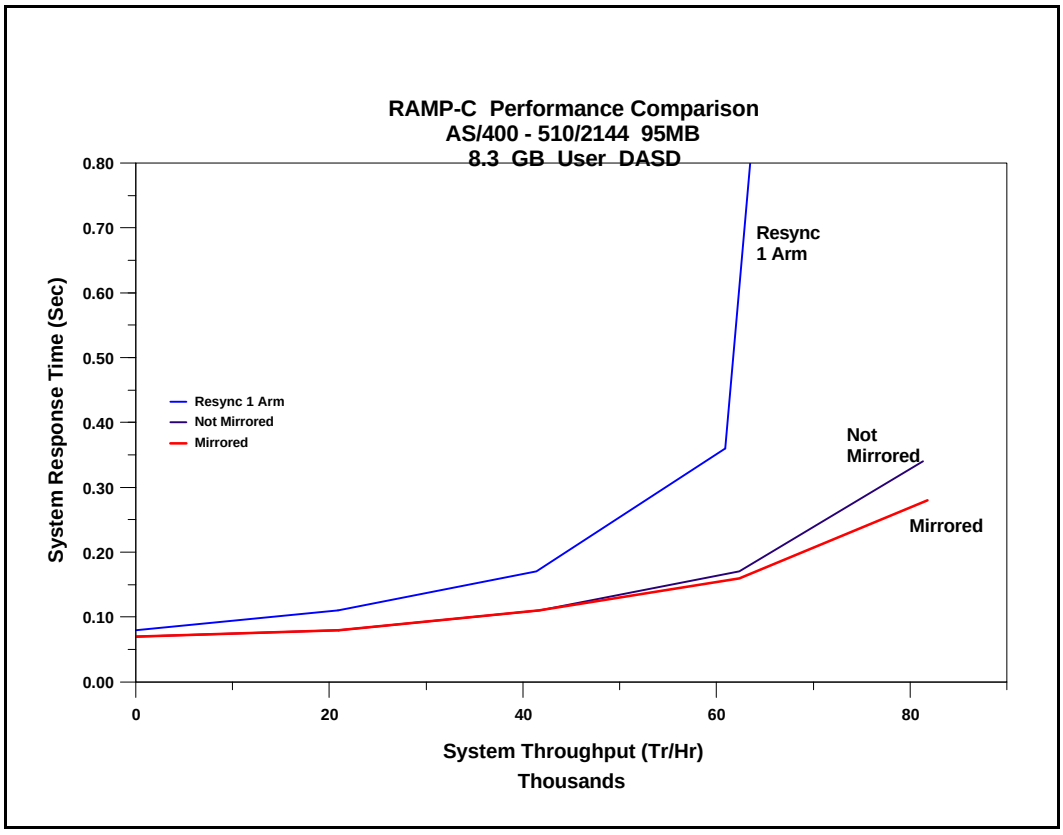



Figure 14.4. System Interactive Performance - Mirrored versus Base - Internal DASD

## Conclusions / Recommendations

- The mirrored configuration provides equal or better interactive performance than the base configuration (not mirrored). The better mirrored performance is due to having more arms to handle the larger number of read ops at higher throughput.

- The system performance is less during the time it takes to resync an arm, especially at higher throughput. The customer could choose to schedule the resync during a period of lower system activity or quiesce some applications during the resync time (20 to 40 minutes for a 1GB device). Larger devices will have proportionally longer resync time.

- This graph is based on RAMP-C workload. Other environments may vary significantly.

## AS/400 System Interactive Performance - RAID

The following graph compares the 65x2 RAID DASD Subsystems with the 9337-580 HA Subsystem. All subsystems contained 8 4GB arms.

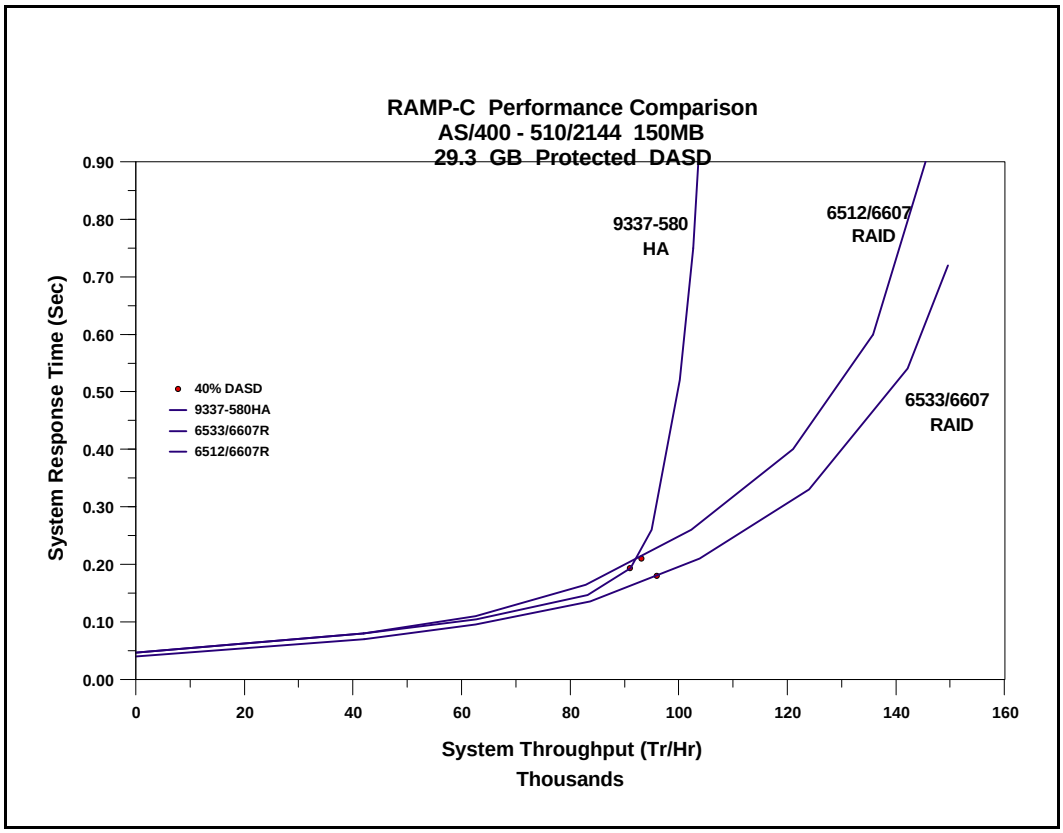



Figure 14.5. System Interactive Performance - RAID Mode

## Conclusions / Recommendations

- The 6533/6607 RAID DASD provides better interactive performance than the 6512/6607 RAID DASD. The 6754, 6751, 6532, 2726, 2740 and 2741 DASD IOP/IOAs have performance characteristics similar to the 6533 IOP over typical operating ranges.

- Performance with the 9337-580 HA is comparable with the 6512/6607 RAID DASD for similar throughput.

- The 9337 measurements were done with 4 parity arms per array and the 65x2 measurements were done with 8 parity arms per array. In general 8 parity arms per array will provide better performance at higher throughputs. At low to medium throughput, there is little performance difference between 4 and 8 parity arms per an 8 arm array. On the 65x2 (also 2726, 2740, 2741, 6533, 6751 and 6754), parity arrays of 8 or more arms should be configured with 8 parity arms if possible.

## Impact of failed DASD in RAID Subsystem

This is a general discussion of RAID-5.

- 65x2 (also 2726, 2740, 2741, 6533, 6751 and 6754) RAID and 9337 HA DASD subsystems let the AS/400 system continue to operate even after a single DASD failure.

- With system checksum, a DASD failure will cause the AS/400 system to stop and an IPL will be required.

- RAID-5 overhead can become significant when a DASD fails.

  **READ**

  - To read from a failed DASD, RAID-5 must read ALL remaining arms in the set. (This means anywhere from 3 to 9 overlapped reads, where 1 was sufficient before). This will have a significant effect on the **failed** DASD subsystem throughput and response time. This degraded mode will last until the DASD is repaired and the "rebuild" of the failed DASD's parity stripes are complete.
  - Reads to other DASD on the same subsystem are unaffected.
  - Bottom line, if the parity array has 4 arms, this results in 1.5 times increase in DASD IO read volume to this array. If the array has 8 arms, the result is 1.75 times increase in DASD IO read volumes to this array.

  **WRITE**

  - There are 3 separate scenarios that apply to RAID-5 writes with one failed DASD.
    - If the failed DASD is not involved (either for data or for the checksum stripe), the writes are handled as normal RAID-5 writes. (2 reads plus 2 writes)
    - With a write to a failed DASD, all remaining DASD in the set must be read and then one write will be done to the checksum stripe. (N-1 reads plus 1 write, N = number of DASD arms)
    - If the DASD that contains the checksum stripe is the failed DASD, then all that is required is a write to the DASD that contains your data. (1 write)
    - Bottom line, if the parity array has 4 arms, each write averages a 3.25 increase in DASD IO write volume to this array. If the array has 8 arms, the result is a 4.13 times increase in DASD IO write volumes to this array.

  **General discussion**

  - When running in "exposed" mode, the fewer the number of arms in each parity array, the smaller the degradation.
  - On systems with smaller amounts of DASD capacity, the degradation will be more noticeable. This is because there are fewer arrays which means that a larger percentage of the DASD operations will be directed to the "exposed" array.
  - The DASD IO to any subsystems that do not have a failed DASD are unaffected.
  - If the Customer cannot tolerate the temporary performance degradation that would occur with a RAID-5 DASD failure, they should consider mirroring.
  - To obtain acceptable performance with a failed RAID-5 DASD, some customers may have to delay nonessential work until after the DASD is repaired. For example, a customer may continue to process their on-line order entries but delay their office tasks.
  - In configurations with small amounts of total DASD space and with high availability requirements, mirroring may be a more satisfactory option
  - The estimated time to rebuild a DASD is approximately 30 minutes for a 8 arm array on a dedicated system with no other jobs running. If other concurrent jobs being run on the system are requesting 130 IOs per second to this DASD subsystem, the rebuild time will increase to approximately 1 hour.

The following chart compares the impact of one failed DASD on several configurations.

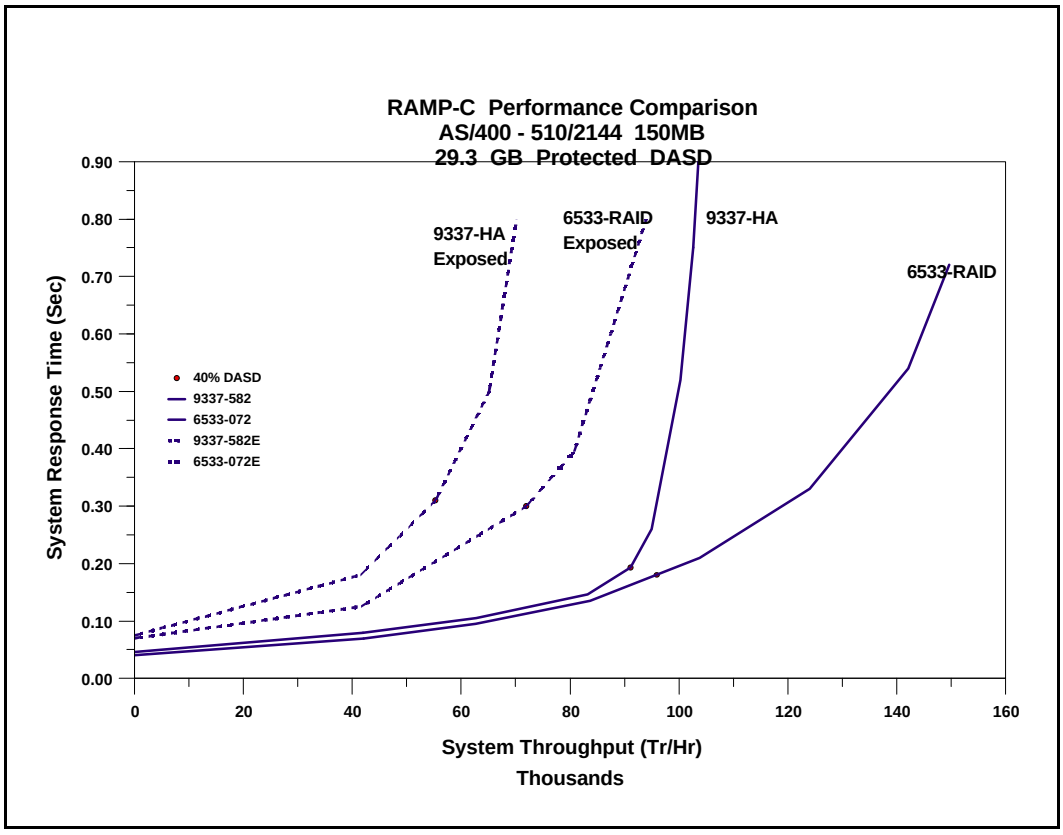



Figure 14.6. Performance Impact of Failed DASD

## Conclusions / Recommendations

- The 6533 performs better in "exposed" mode than the 9337-580 due to the ability of the 6533 to handle higher throughput more efficiently. The 6754, 6751, 6532, 2726, 2740 and 2741 DASD IOP/IOAs have performance characteristics similar to the 6533 IOP over typical operating ranges.

- With a parity array of 8 arms on a 6533, "exposed" mode throughput is about half of normal throughput.

- If there were "n" 65x2s in the configuration, 1/n of the DASD IOs are to the "exposed" 65x2.

- As "n" gets larger, the impact of a DASD failure to overall system performance is reduced.

- Additional degradation occurs during rebuild for both 65x2 and 9337. The rebuild can be scheduled for periods of lower system utilization.

## Ops/Sec/GB Guidelines for DASD Subsystems

The metric used in determining DASD subsystem performance requirements is the number of I/O operations per second per installed GB of DASD (Ops/Sec/GB). Ops/Sec/GB is a measurement of throughput per actuator. Since DASD devices have different capacities per actuator, Ops/Sec/GB is used to normalize throughput for different capacities. An Ops/Sec/GB range has been established for each

DASD type so that if the DASD subsystem performance is within the established range, the average arm percent busy will meet the guideline of not exceeding 40%.

The implementation of the 4KB page size on RISC will improve system DASD IO efficiency. As a result of the larger page size, some DASD subsystem interactive Ops/Sec/GB ranges will appear lower than IMPI.

The following bar charts show the "rule of thumb" for the Physical system Ops/Sec/GB of usable space that internal DASD subsystems can achieve with various DASD types. (To compute usable GB, we assume that the DASD subsystems have 8 disk units installed). The top of each bar is the volume of 7K transfer, 80% 1/3 seek, 30% write operations that each model can achieve when it is 40% busy. For the 6502, we assume that the write cache has an efficiency of 55%. For the 6512, 6532,6533, 6751, 6754, 2726, 2740 and 2741 we assume that the write cache has an efficiency of 65%. The vertical scale is the volume of physical Ops issued from the system. The 65x2 RAID disk subsystem bars are lower because of the additional work that these subsystems must do to maintain the RAID-5 parity stripes.

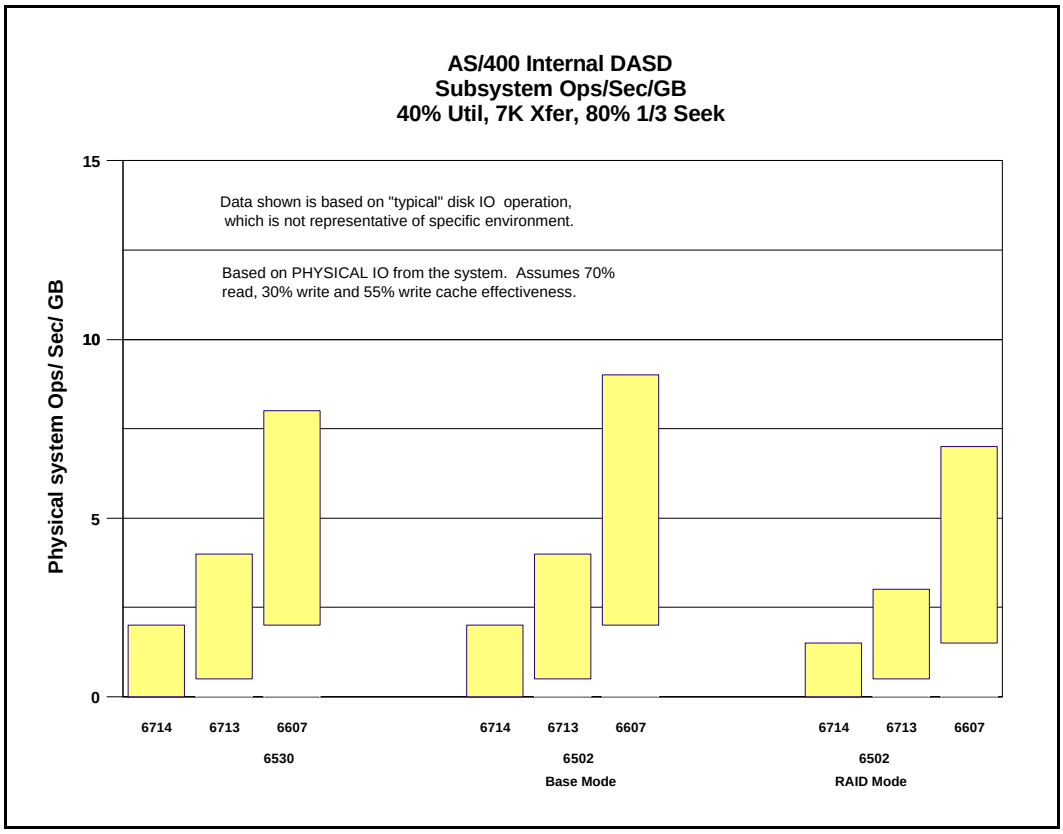



Figure 14.7. Ops/Sec/GB - Internal DASD

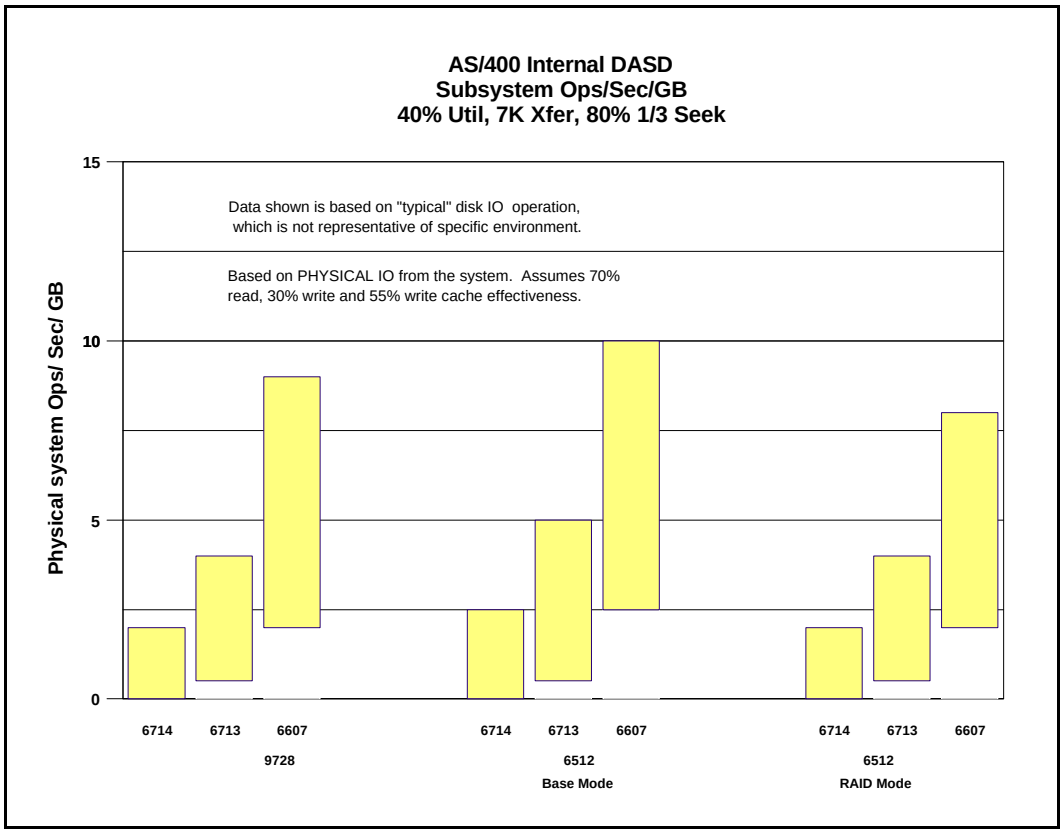



Figure 14.8. Ops/Sec/GB - Internal DASD (continued)

## Conclusions / Recommendations

- In general, the higher the capacity of the DASD device the lower it's throughput range will be. The 6714 device (17548 MB per arm) has a lower throughput range than the 6606 device (1967 MB per arm).

- Typically, DASD subsystems will have a lower throughput range when operated in RAID mode rather than Base mode. The throughput difference due to RAID will tend to be smaller for workloads characterized by higher read to write ratios.

- The 6714 DASD (17548 MB per arm) is more appropriate when the capacity requirement is very large and the Ops/Sec/GB requirement is less than 3.

- The 6713 DASD (8589 MB per arm) is more appropriate when the capacity requirement is large and the Ops/Sec/GB requirement is less than 4.

- The 6607 models (4194 MB per arm) will be the appropriate choice for almost all other situations. *NOTE: The 6606 models (1976 MB per arm) may be needed for extreme cases where the capacity requirement is very low relative to the system disk ops/sec rate. Refer to section OSGB for a discussion of the performance limits for each DASD type. 6606 is only needed for cases exceeding the limits for 6607.

- The low cost 9728 DASD IOA provides a similar throughput range when compared to the 6530 DASD IOP.

- The 6512 DASD subsystem provides an improved throughput operating range when compared with the 6502 DASD subsystem.

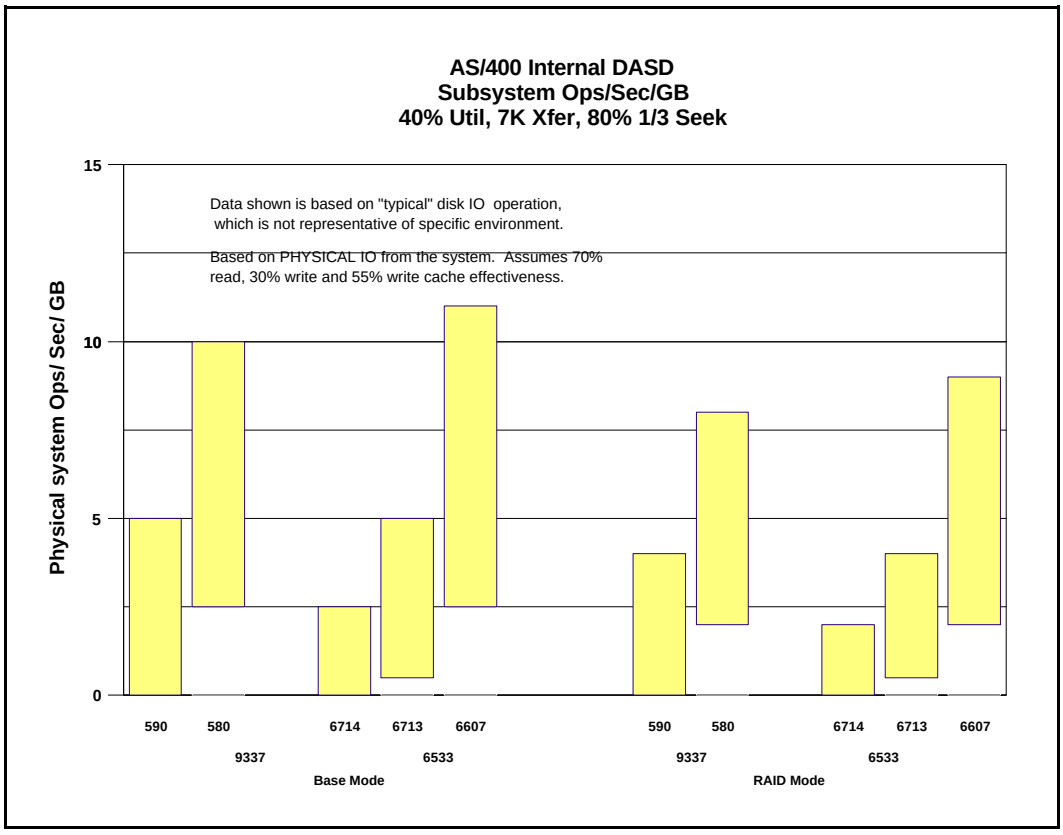



Figure 14.9. Ops/Sec/GB - Internal versus External DASD

## Conclusions / Recommendations

- The 6533 DASD subsystem provides an improved throughput operating range when compared with the 6512 DASD subsystem. The 6754, 6751, 6532, 2726, 2740 and 2741 DASD IOP/IOAs have performance characteristics similar to the 6533 IOP over typical operating ranges.

- The 6533 DASD subsystem provides an improved throughput operating range when compared with the 9337-5xx DASD subsystem.

- The 6533 DASD IOP provides a better throughput range when compared to the low cost 9728 DASD IOA.

## Using the OPS/SEC/GB Chart

The Ops/Sec/GB chart above should be used as a guideline on what DASD model is the appropriate choice when adding or upgrading DASD. In conjunction with this chart, you should utilize results obtained from the Performance Tools LPP report to determine what DASD model meets your DASD performance

requirements. For more detailed DASD performance analysis, it is recommended to use BEST/1-400, the capacity planner for the AS/400.

Operations per second is a measurement of throughput per actuator. Since DASD devices have different capacities per actuator, operations per second per GB is used to normalize throughput for different capacities. To determine the operations per second of your current operating environment, follow the procedure outlined below. The value obtained by this procedure will help determine what DASD model will meet your current or projected DASD performance requirements.

1. Collect performance data using the Performance Monitor. Be sure to collect this data during peak activity for at least a one hour time period using 10 minute sample intervals.

2. Print the Performance Tools System Report using the PRTSYSRPT command. Then, refer to the "Disk Utilization" section of the Performance Tools LPP System Report. From this report the following data can be obtained:

   - Total operations per second - Use Op Per Second column
   - Total GBs of DASD installed - use Size (M) column

3. To determine the total GBs installed, simply add the "Size (M)" column and divide by 1000. When adding the total GBs, you should ONLY include the disk units you plan to replace. Also, if Mirroring is active, divide the total GB being mirrored by 2 when calculating the sum.

4. To determine the total operations per second, add the total operations per second number ("Op Per Second" column). When adding the total operations per second, you should ONLY include the disk units you plan to replace. Also, if mirroring is active, you need to divide the total number of operations per second for all mirrored units by 2.

5. To determine the operations per second per GB, divide the total operations per second you calculated in step 4, by the total GBs installed value you calculated in step 3.

You can then use the operations per second value to determine what model of DASD best fits your current or projected DASD performance requirement.

**EXAMPLE**

If you wanted RAID mode, you could select 6533/6607 models if the physical system IO/Sec/GB were at 11 or fewer.

---

## 14.3 DASD Performance - Batch

**Commercial Batch - Base versus RAID**

This section shows the results of running one of IBM's batch workloads in a dedicated environment. The workload performs the following functions :

- Sequential and Keyed Record Copy
- Sequential and Keyed Program Read
- Sequential and Keyed Record Read/Update
- Record Matching
- Adding and Removing Members
- RGZPFM of 500,000 Records
- Average - 40% Read Ops, 60% Write Ops, 17 KB/IO
- 70% Synchronous / 30% Asynchronous Ops

The workload was run in a 24 MB memory pool. A 6533-18GB was compared to an equivalently configured 6532-18GB with RAID protection turned on and off.

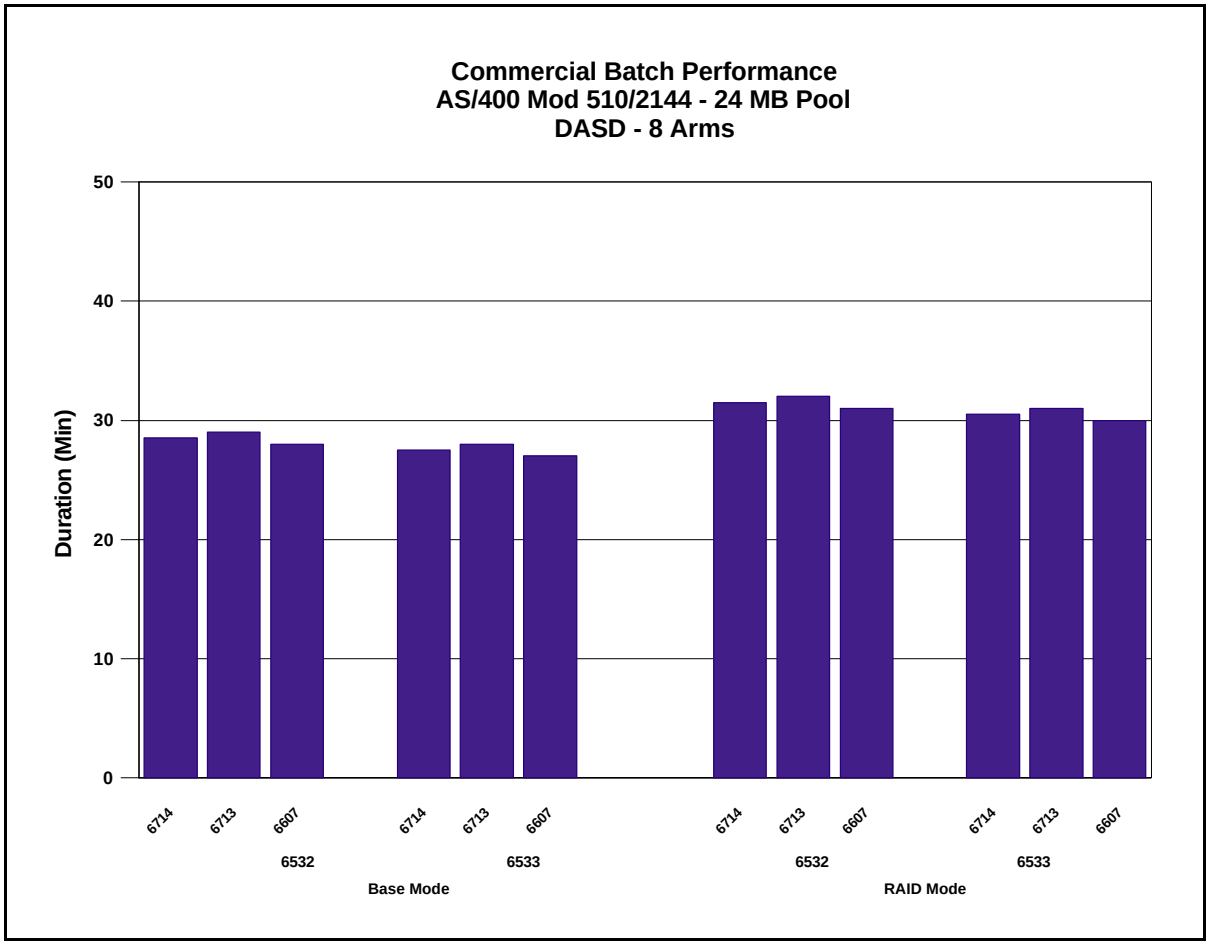



Figure 14.10. Commercial Batch Performance - Base versus RAID

**Conclusions / Recommendations**

- The 6533-18GB (6714) has better performance than the 6532-18GB (6614) for this commercial batch workload.

- The 6714 DASD has slightly better performance than the 6713 DASD.

The workload was run in a 24 MB memory pool. A 6512-8GB was compared to an equivalently configured 9337-590 with RAID protection turned on and off.

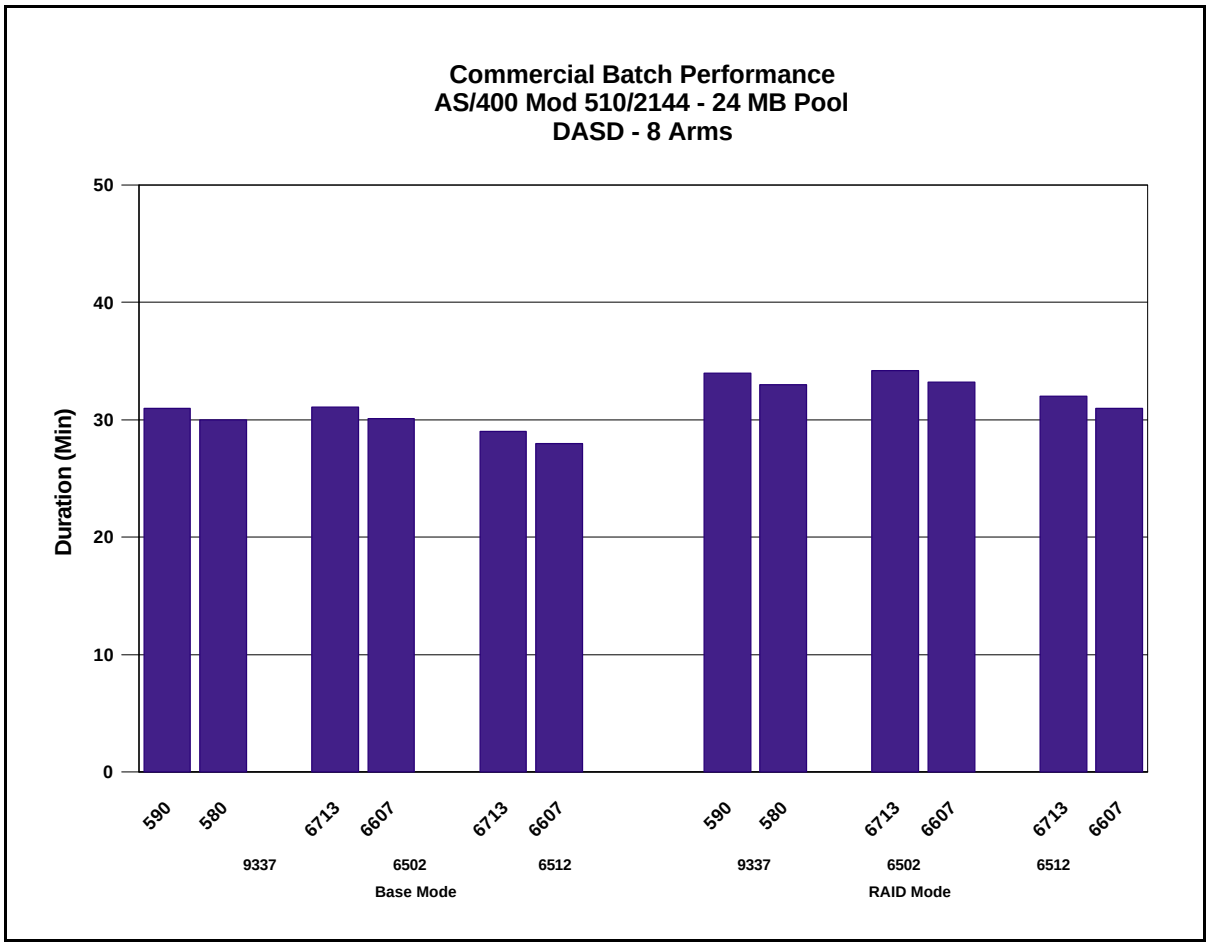



Figure 14.11. Commercial Batch Performance - Base versus RAID

## Conclusions / Recommendations

- The 6512-8GB (6613) has better performance than the 6502-8GB (6613) for this commercial batch workload.
- The 6512-8GB (6613) has better performance than the 9337-590.
- The 6502-4GB (6607) has similar performance as the 9337-580 for this commercial batch workload.
- The 6607 DASD has slightly better performance than the 6713 DASD.

## 14.4 DASD Performance - General

### Mixing RAID DASD with other DASD in one ASP

Combining 65x2 (also 2726, 2740, 2741, 6533, 6751 and 6754) RAID DASD with mirrored DASD in a single ASP is allowed. Combining RAID DASD with mirrored DASD on the same 65x2 is also allowed.

### Write Intensive Applications (eg RESTORE)

RAID-5 (like system checksum) can have a significant impact on batch type programs that issue many writes in a short period of time. This is due to the four times increase in disk IO required for each write. The 65x2 write cache handles this impact for almost all scenarios except those that write hundreds of writes to the DASD in a very short period of time. Even in this worst case scenario, with only one 65x2 array configured, the restore of a large file took only 30% longer than a restore to a standard" 6530 (or 9728) configuration.

The 65x2 can restore small objects faster than 6530 because of the 65x2 write cache. The write cache provides fast completion of write requests and is able to "stay ahead" of the system.

The 65x2 RAID models offer significant advantages in availability, reliability, price, etc. One of the "costs" of the availability advantage is the increased time to restore data. This increase in time needs to be considered when planning the installation of RAID-5 disk. The time to load data onto the RAID-5 boxes must be included in the overall installation planning. With the increased availability and reliability offered by 65x2 RAID, the necessity to reload the data again due to a single disk unit failure will be eliminated.

### DST "Add Unit"

Part of the process of adding DASD to a system is using the Dedicated Service Tool (DST) to "Add Unit". This ensures that the entire DASD(s) is initialized with a X'00' data pattern and verified.

When multiple DASD are added at once, the system will add up to 16 units in parallel.

The time for adding up to 16 units on a DASD IOP (Base mode) is approximately

- 48 minutes for 2 GB arms (6606)
- 86 minutes for 4 GB arms (6607)
- 162 minutes for 9 GB arms (6713)
- 302 minutes for 18 GB arms (6714)

The Dedicated Service Tool is also used to start and stop parity (RAID-5) arrays on the 65x2 (also 2726, 2740, 2741, 6751 and 6754) IOP. When a parity array is initially set up, the fastest approach is to start parity on an array first and then add the arms to an ASP. The time required for this process (start parity and add) on two 8 arm arrays is approximately:

- 48 minutes for 2 GB arms (6606)
- 86 minutes for 4 GB arms (6607)
- 162 minutes for 9 GB arms (6713)
- 302 minutes for 18 GB arms (6714)

If the arms are added to the ASP before starting the array, then the time required may double. If a system IPL occurs between starting the array and adding the arms to an ASP, then the time required could be 3 times as long.

If the arms are currently part of an ASP, then starting an array will take longer because the system may need to move data before it synchronizes the parity stripes. This could take up to:

- 90 minutes for 2 GB arms (6606)
- 160 minutes for 4 GB arms (6607)
- 300 minutes for 9 GB arms (6713)
- 580 minutes for 18 GB arms (6714)

Stopping parity on an 8 arm array takes about:

- 70 seconds for 2 GB arms (6606)
- 120 seconds for 4 GB arms (6607)
- 220 seconds for 9 GB arms (6713)
- 410 seconds for 18 GB arms (6714)

## 14.5 Integrated Hardware Disk Compression (IHDC)

Integrated Hardware Disk Compression (IHDC) is a new DASD capability for V4R3. IHDC has the following characteristics :

- Data is dynamically compressed/decompressed by the DASD subsystem controller (IOP/IOA) independent of the AS/400 system processor
- Compressed data is not seen above the DASD controller level
- Compression is performed by an LZ1 compression chip on the DASD controller

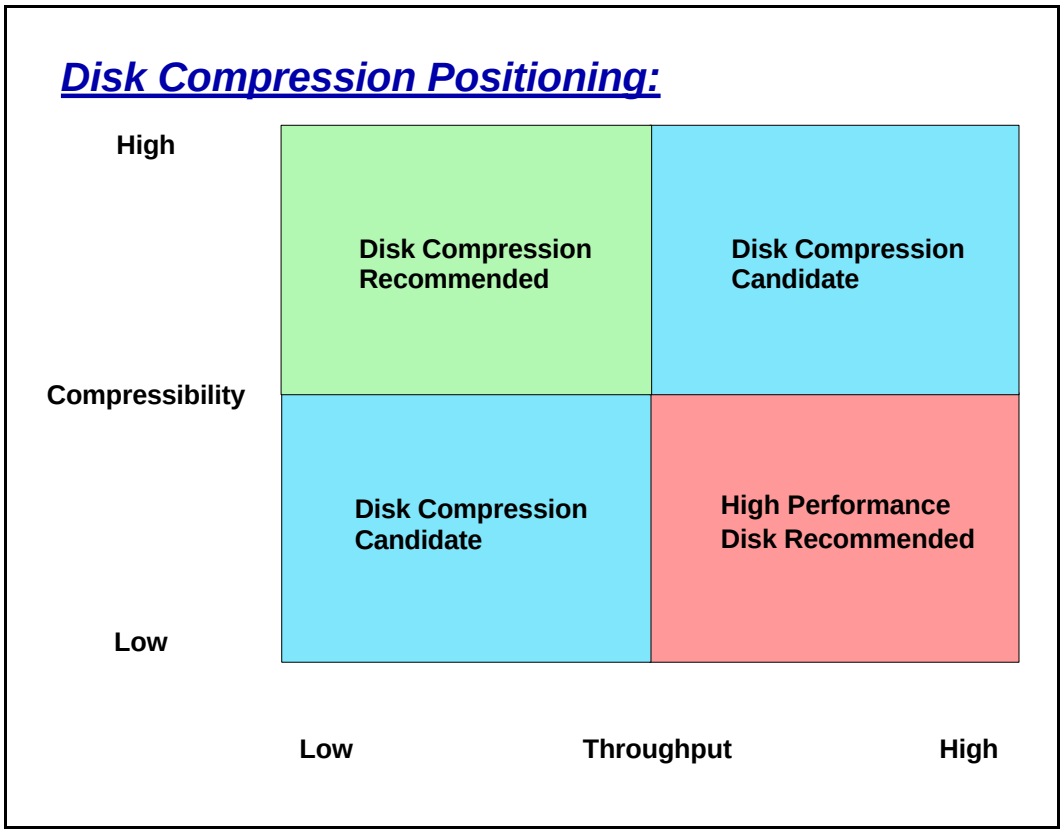
- An average 2X compression ratio, with up to 4X achievable (data dependent)
- Customer on/off option provided at disk arm level
- RAID and mirroring is supported (no additional restrictions)
- With compression, AS/400 disk capacity maximums can be exceeded.

IHDC provides the following customer value :

- Reduces cost of on-line storage
- Provides better performance than typical software data compression
- Enables new applications
- Protects investment - ability to increase capacity of installed DASD
- Provides a storage management solution when used with hierarchical storage management (HSM).

IHDC has the following requirements :

- Requires a compression enabled IOP/IOA
- Compressed DASD must be configured in user ASPs only
- 17.54GB disks are supported with V4R4 and future releases
- Disks must be unconfigured to be enabled for compression
- Data must be saved before compression is disabled to avoid loss
- Compressed disks can be migrated only to compression enabled IOP/IOAs.




- High compressibility, low throughput
  - ❖ The disk space reduction benefit of compression will be realized, and both reads and writes to compressed disk should outperform uncompressed disk
  - ❖ Disk compression is recommended
- Low compressibility, low throughput
  - ❖ The lower the compressibility, the smaller the disk space reduction benefit, although disk performance will likely not suffer
  - ❖ Disk compression is a candidate if compressibility is high enough
- High compressibility, high throughput
  - ❖ The disk space reduction benefit of compression will be realized, and compressed disk performance may be close to uncompressed disk performance, or even faster for read intensive applications
  - ❖ Disk compression is a candidate, especially for read intensive applications
- Low compressibility, high throughput

- ❖ The disk space reduction benefit of compression will be minimal, and performance will likely suffer especially for write intensive applications

- ❖ High performance disk (uncompressed) is recommended

**DASD Compression Performance Guidelines**

Precise performance projections for IHDC are not possible due to :

- Compressibility of data which can vary greatly
- Workload characteristics of specific applications

DASD compression may cause performance to vary, in general :

- System performance impacts will be minimal when DASD operations are light
- For data with low compression rates (< 2X), DASD read/write performance will generally be slower than for uncompressed data
- For data with high compression rates (> 3X), DASD read/write performance can be faster than for uncompressed data
- DASD Read intensive workloads will typically perform better than DASD Write intensive workloads
- Interactive applications with a mixture of DASD reads and writes with medium to heavy DASD operations should use high performance uncompressed DASD
- With compressed disks it is critical that they operate within a reasonable margin below 'dasd full', otherwise performance will be greatly affected. In contrast to uncompressed disks, when a compressed disk approaches full (approximately 85%) a disk defragmentation task is started within the IOP/IOA to recover fragmented storage and may take considerable time to finish. Since this task runs concurrently with system operations, performance will be degraded until this task completes.
- Mixing Compressed DASD with Uncompressed DASD on the same IOP/IOA may impact the performance of the Uncompressed DASD due to higher IOP/IOA utilization.
- Mixing Compressed DASD with Uncompressed DASD within the same user ASP is supported but is not recommended due to potential performance impact caused by unbalanced disk utilization.

DASD compression is intended for :

- Vast amounts of historical or archive data
- Low activity data
- Spool files

- Journals
- Save files (staging)

Not for highly volatile data or data already compressed (images, etc.)

Types of applications that can benefit from DASD compression :

- Data warehouse, data mining
- On-line access to archive data
- On-line viewing of reports (paper or micro-fiche replacement)
- Part of a hierarchical storage management strategy

Applications are a candidate for DASD compression if :

- Additional DASD storage is required
- Application data can be partitioned (at least partially) into user ASPs
- Top application performance is not required

Refer to *AS/400 Backup and Recovery V4R3* (SC41-5304-02) for more information about configuring and using compressed DASD.

**Interactive Performance with Compressed DASD**

The following graph compares relative interactive system performance of an AS/400 model 640/2239 configured with 16-arm user ASPs of Compressed, Uncompressed, RAID/Compressed and RAID/Uncompressed DASD. The graph compares the performance results of running the RAMP-C workload in each of the 4 user ASPs.

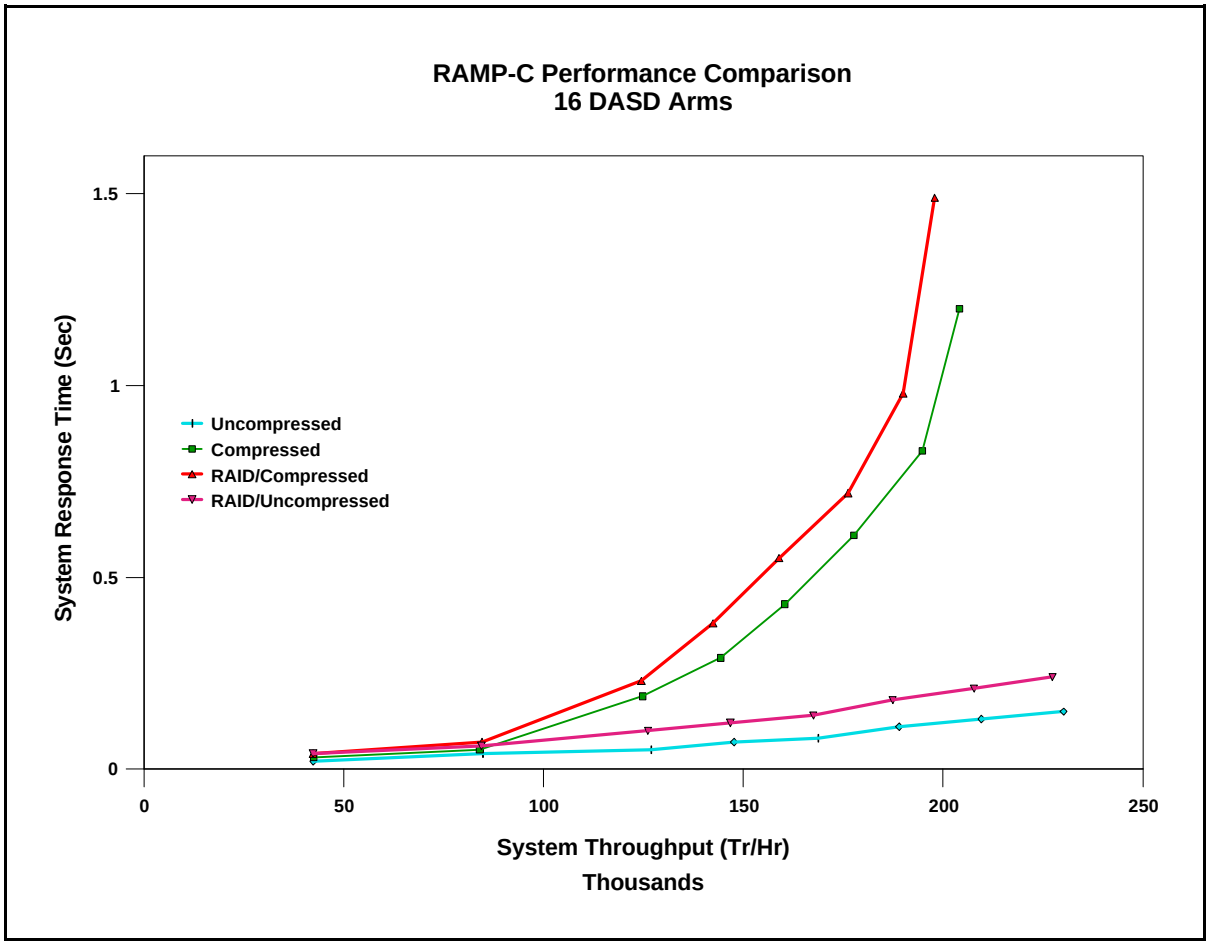



Figure 14.13. System Interactive Performance - Compressed DASD

## Conclusions / Recommendations

- At lower throughputs and with equal number of disk arms, Compressed DASD has similar (0-10% degradation) system performance characteristics to Uncompressed DASD for interactive workloads.
- Compressed DASD is not appropriate for high throughput (ie. write intensive) environments.
- Compressed DASD performance can often be improved by maintaining :
  - ❖ Disk CPU utilization below 60% (about 360 ops/sec per IOP/IOA)
  - ❖ Disk utilization below 40% (about 23 ops/sec per disk arm)
- For the above graph, the Compressed disk 40% utilization point is at 146,000 transactions per hour and the RAID/Compressed disk 40% utilization point is at 140,000 transactions per hour.
- For the above graph, the Compressed Disk CPU (IOP/IOA) 60% utilization point is at 144,000 transactions per hour and the RAID/Compressed Disk CPU 60% utilization point is at 138,000

transactions per hour. Above these limits, system performance with Compressed DASD tends to degrade noticeably as the throughput increases.

- Compressed DASD with RAID has similar system performance characteristics to Compressed DASD without RAID at lower throughputs. At higher throughputs, RAID/Compressed DASD performance is less due to higher utilization's for the same op rates. The same criteria as above should be followed for obtaining acceptable RAID/Compressed DASD performance.

- Just as with uncompressed DASD, the number of disk arms must be adequate to support anticipated op rates.

- Configuring fewer disk arms per IOP/IOA will typically improve the performance of Compressed DASD. DASD subsystems with 8 arms per IOP/IOA will usually perform much better than those with 16 arms per IOP/IOA.

**Batch Performance with Compressed DASD**

The following chart compares system performance of various batch type applications while running on an AS/400 model S30/2259 configured with 16-arm user ASPs of Compressed, Uncompressed, RAID/Compressed and RAID/Uncompressed DASD. Batch run time was measured in each of the 4 user ASPs for 7 batch tests with the following DASD I/O characteristics :

1. Sequential read ops, 5 KB/op, OS/400 Expert Cache off

2. Sequential read ops, 60 KB/op, OS/400 Expert Cache on

3. Sequential read and write ops, 68% reads, 5 KB/op, OS/400 Expert Cache off

4. Sequential read and write ops, 17% reads, 50 KB/read op, 5 KB/write op, OS/400 Expert Cache on

5. Random read ops, 7 KB/op, OS/400 Expert Cache off

6. Random write ops, 8 KB/op, OS/400 Expert Cache off

7. Sequential read and write ops, 14% reads, 5 KB/op, OS/400 Expert Cache off

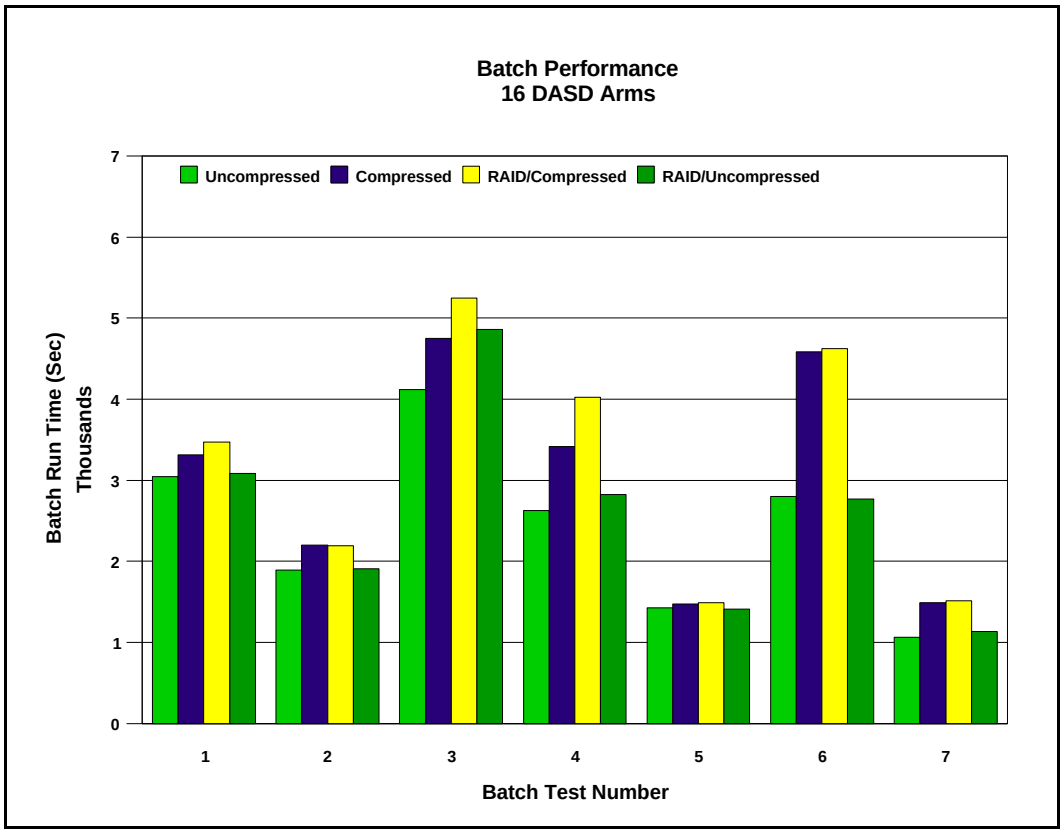



Figure 14.14. Batch Run Time Performance - Compressed DASD

## Conclusions / Recommendations

- For batch applications characterized by DASD read ops, system performance varied only slightly between Compressed, Uncompressed, RAID/Compressed and RAID/Uncompressed DASD ASPs.

- For batch applications characterized by DASD write ops, system performance was slower for Compressed and RAID/Compressed than for Uncompressed and RAID/Uncompressed DASD ASPs.

- For batch applications characterized by a mixture of DASD read and write ops, system performance was slower for Compressed and RAID/Compressed than for Uncompressed and RAID/Uncompressed DASD ASPs. The magnitude of the performance difference typically depends on the percentage of write ops.

- OS/400 Expert Cache provided better batch system performance when active.

## Save/Restore Performance with Compressed DASD

The following charts compare system performance of Save/Restore operations while running on an AS/400 model S30/2259 configured with 16-arm user ASPs of Compressed, Uncompressed, RAID/Compressed and RAID/Uncompressed DASD. The System ASP (ASP1) was configured with Uncompressed DASD. Data transfer rates were measured in each of the 4 user ASPs for 6 different types of Save/Restore tests :

1. SAV to a 3590 tape from the user ASP (Read data from Compressed DASD)
2. RST from a 3590 tape to the user ASP (Write data to Compressed DASD)
3. SAV from ASP1 to a *SAVF on the user ASP (Write data to Compressed DASD)
4. RST from a *SAVF on the user ASP to ASP1 (Read data from Compressed DASD)
5. SAV to a *SAVF on ASP1 from the user ASP (Read data from Compressed DASD)
6. RST from a *SAVF on ASP1 to the user ASP (Write data to Compressed DASD)

The data used for the first chart has a compression ratio of 2X and the data for the second chart has a compression ratio of 4X.

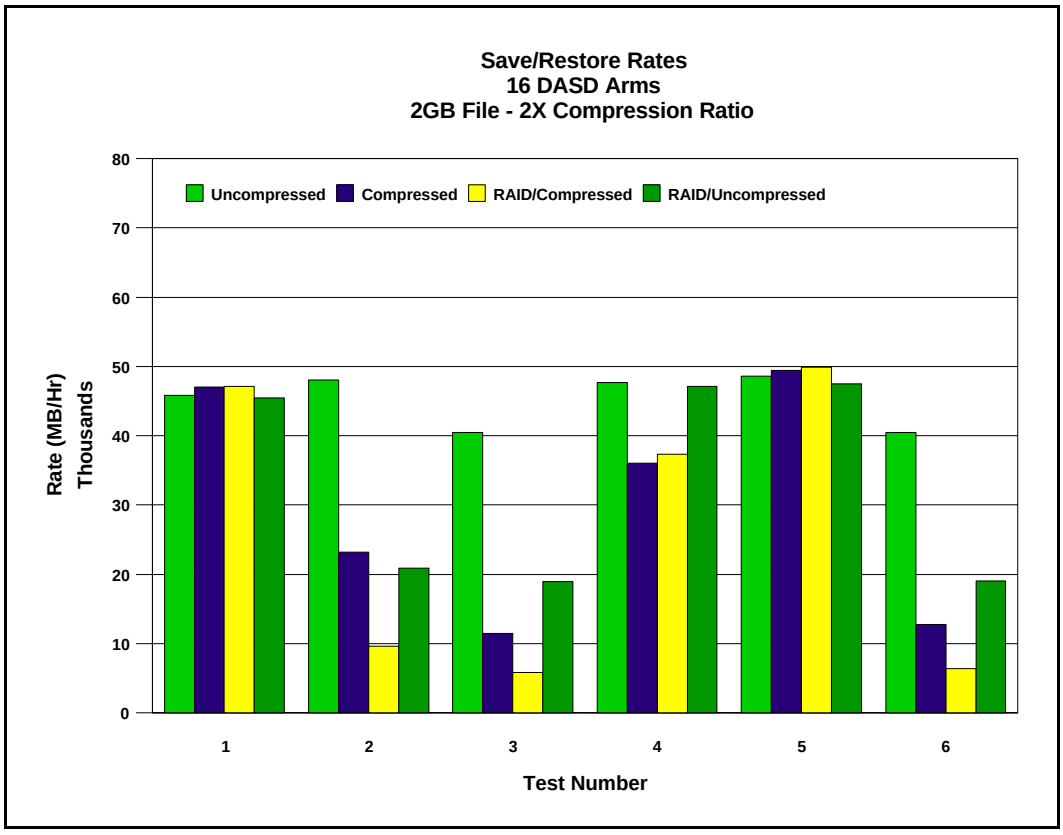



Figure 14.15. Compressed DASD Save/Restore Rates - 2X Compressibility

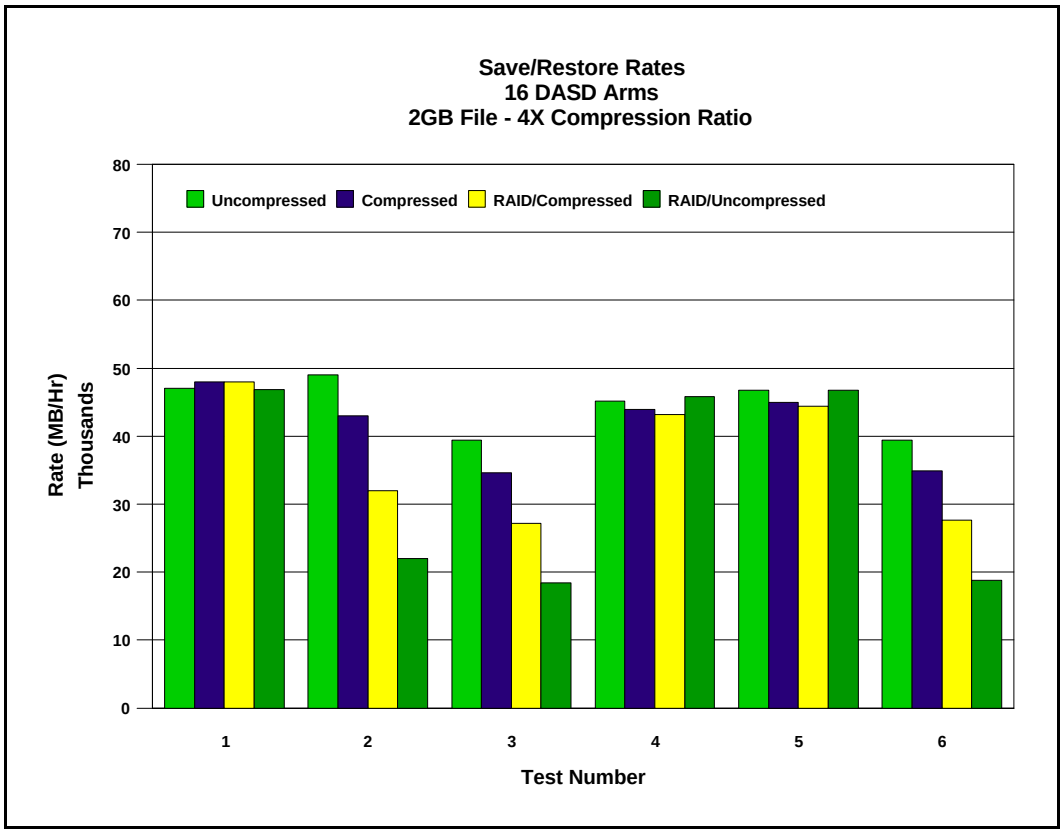



Figure 14.16. Compressed DASD Save/Restore Rates - 4X Compressibility

## Conclusions / Recommendations

- For Save operations to tape, Compressed DASD (also RAID/Compressed DASD) has approximately the same system performance characteristics as Uncompressed DASD. Save operations primarily issue read op to DASD and read op performance is very similar for Compressed, RAID/Compressed and Uncompressed DASD.

- For Restore operations from tape, Compressed DASD performance is highly dependent upon the compressibility of the data being restored - the better the data compresses the better the restore performance.

  - ❖ Performance can range from 50% degradation (2x compression) to 10% degradation (4x compression) for Compressed DASD compared to Uncompressed DASD.

  - ❖ Performance can range from 50% degradation (2x compression) to 40% improvement (4x compression) for RAID/Compressed DASD compared to RAID DASD. The improvement is due to better write cache efficiency and smaller ops because of data compression.

  - ❖ Restore operations primarily issue large write ops to DASD along with allocate ops. Performance degradation occurs because ops have higher overhead due to compression and generate higher utilization for the IOP/IOA and devices.

## Data Migration Performance with Compressed DASD

The following chart compare system performance of BRMS data migration operations while running on an AS/400 model S30/2259 configured with 16-arm user ASPs of Compressed and Uncompressed DASD. The System ASP (ASP1) was configured with Uncompressed DASD. Data transfer rates were measured between ASP1 and each of the 2 user ASPs for 4 different types of data :

- Large data base file
- Typical user mix of data
- Source data
- DLO data

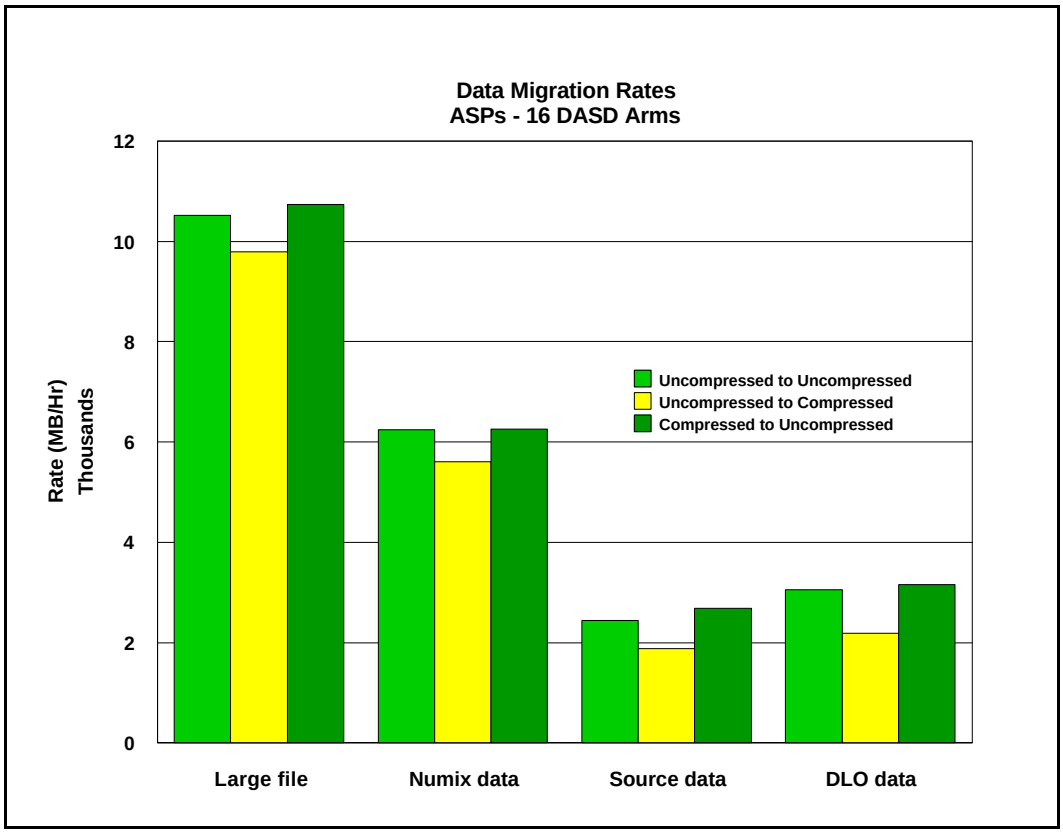



Figure 14.17. Compressed DASD Data Migration Rates

## Conclusions / Recommendations

- Migrating data from an Uncompressed ASP to a Compressed ASP is usually slower than migrating data between two Uncompressed ASPs. The magnitude of the performance difference typically depends on the type and compressibility of the data being migrated, the lower the compressibility, the slower the transfer rate.

- Migrating data from a Compressed ASP to an Uncompressed ASP is usually as fast as or slightly faster than migrating data between two Uncompressed ASPs.

## Data Throughput Performance with Compressed DASD

The following graph compares relative data throughput performance of Compressed and Uncompressed DASD read and write operations for the range of data compression ratios supported. This graph is intended to show in general how data throughput for Compressed DASD is dependent upon the compressibility of the data being transferred.

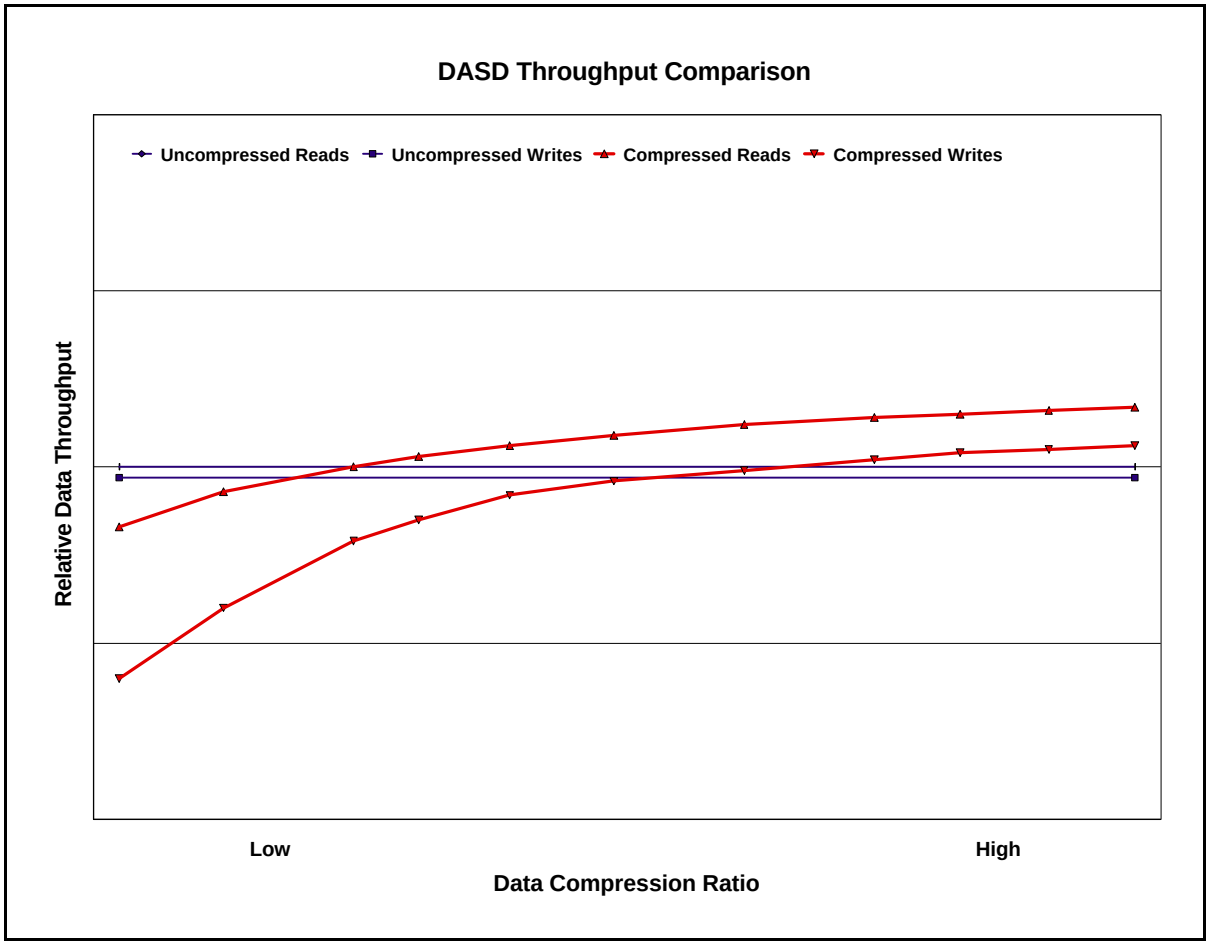



Figure 14.18. Compressed DASD Relative Data Throughput

## Conclusions / Recommendations

- Performance of Uncompressed DASD read and write operations is independent of the compression ratio of the data.

- As compression ratios improve, performance improves accordingly, with read always leading write performance .

- At high compression ratios, compression performance can actually exceed uncompressed performance.

## Response Time Performance with Compressed DASD

The following graph compares relative response time performance of Compressed and Uncompressed DASD read and write operations for the range of data throughput. This graph is intended to show in general how response time for Compressed DASD depends upon the data throughput of the DASD devices.

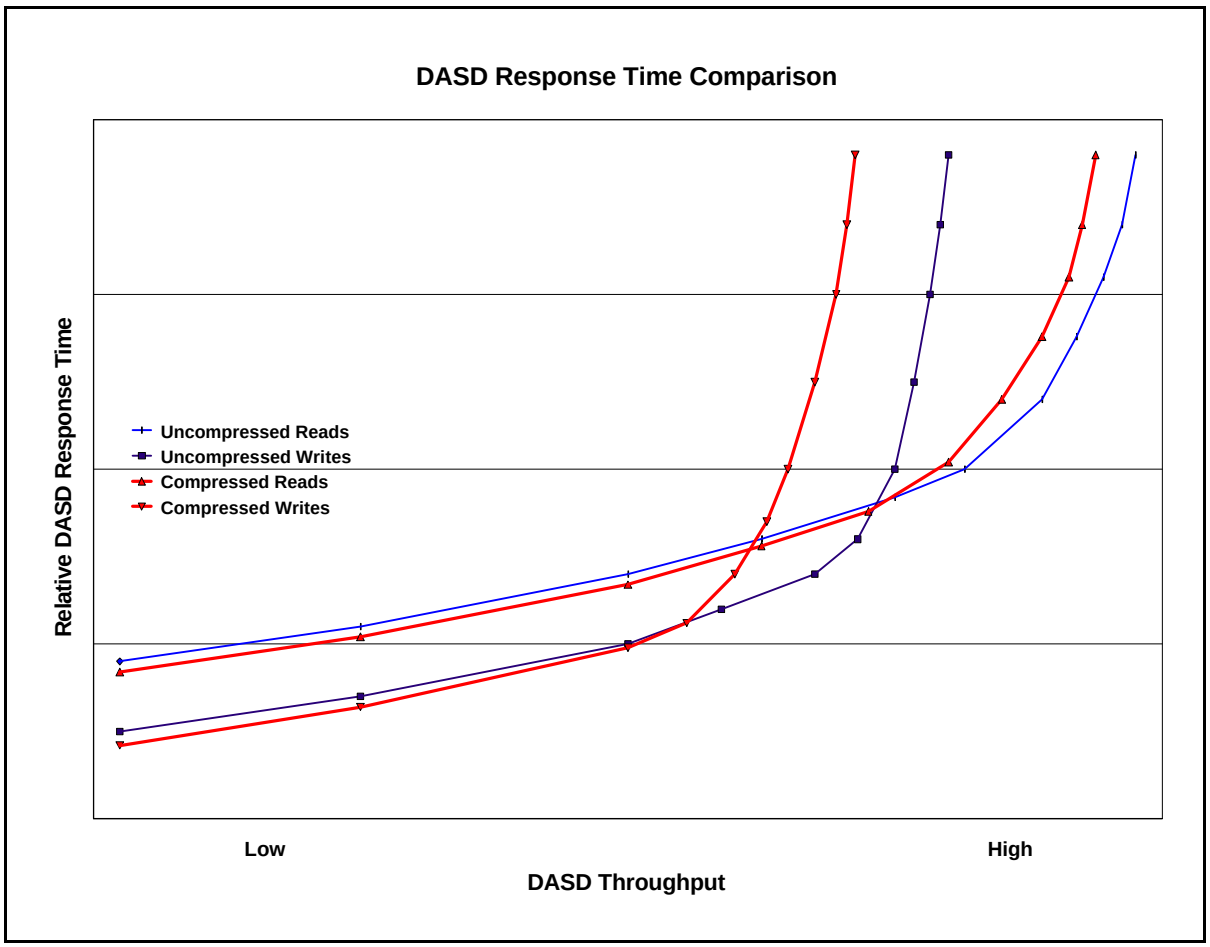



Figure 14.19. Compressed DASD Relative Response Time

## Conclusions / Recommendations

- As data throughput increases, DASD response time increases until it reaches a bottleneck at higher throughputs.
- Compression DASD tends to bottleneck sooner than Uncompressed DASD.
- Writes tend to bottleneck sooner than Reads.

## 14.6 DASD Subsystem Performance Improvements for V4R4

This section discusses the DASD subsystem performance improvements that are new for the V4R4 release. These consist of the following new hardware and software offerings :

- PCI RAID Disk Unit Controller (#2748)
- 10K RPM Disks (#6717)
- Storage/PCI Expansion Tower (#5065)
- Extended Adaptive Cache

The PCI RAID Disk Unit Controller (#2748) is a new DASD IOA that attaches to the system PCI bus. It provides performance improvements by increasing Fast Write Cache to 26 MB (from 4MB) and adding SCSI LVD (Low Voltage Differential Signaling) for SCSI Wide-Ultra2 (80MB) support on a new storage adapter. When the 2748 IOA is configured for DASD Compression the Fast Write Cache is limited to 4 MB.

The 6717 is a new 10K RPM Disk (9GB) that provides faster data access than the previous 7200 RPM devices. It can be attached only with 6532, 6533, 6751, 6754, 2726, 2740, 2741, 2748 and 2763 IOP/IOAs. It can be used as a load source and can be RAIDED and MIRRORED with its 7200 RPM counterparts.

The Storage/PCI Expansion Tower (#5065) provides connectability of the new PCI RAID Disk Unit Controller and 10K RPM Disks to a system which has SPD buses only.

The Extended Adaptive Cache is a feature of the new PCI RAID Disk Unit Controller that provides improved performance characteristics especially in read intensive workloads. The Extended Adaptive Cache requires a Read Cache Device (#4331 or #6831) for memory. For V4R4, the Read Cache Device is a 1.6GB volatile solid state disk. The Extended Adaptive Cache is managed such that ranges of data actively being read are brought into and kept in the cache for as long as they remain active. The goal is to improve performance for read-only or read-write commercial type workloads, while not harming the performance of write-only, random, sequential read, or sequential write workloads.

Extended Adaptive Cache was created to complement other caches within the system and designed to meet the specific needs of AS/400 system users. Although Extended Adaptive Cache functions independently from Expert Cache (which uses main memory), the DASD IOA fast write cache, and device read-ahead buffers, it takes each caching strategy into account as it tracks the physical I/O activity. NOTE: DASD Compression and Extended Adaptive Cache are mutually exclusive.

Although Extended Adaptive Cache has proven to be highly effective in improving performance on many types of workloads, the cache effectiveness is workload dependent. Both the system configuration and type of I/O activity have a direct impact on the performance benefits of Extended Adaptive Cache. Therefore the Extended Adaptive Cache Simulator was created to enable AS/400 system users to know what benefits would be realized by adding Extended Adaptive Cache to their system, before having to purchase the actual cache memory. Extended Adaptive Cache Simulator is an integrated

performance evaluation tool that uses the same algorithms that would manage Extended Adaptive Cache if it were activated.

Refer to the Extended Adaptive Cache white paper at http://www.as400.ibm.com/hsmcomp/EACacheWhitepaper.htm for additional information.

## DASD Subsystem Performance - PCI RAID Controller (#2748) and 10K RPM Disks

The following bar graph compares the service times for the new AS/400 DASD subsystem offerings. The new 2748 IOA is compared to the previous 2741 IOA and the new 10K RPM 6717 disk is compared the previous 7200 RPM 6607, 6713 and 6714 disks. The IO operations being performed are 7KB transfer size, 70% are reads and 30% are writes, and 80% require a seek over 1/3 of the disk surface while 20% require no seek. Queuing time is not included.

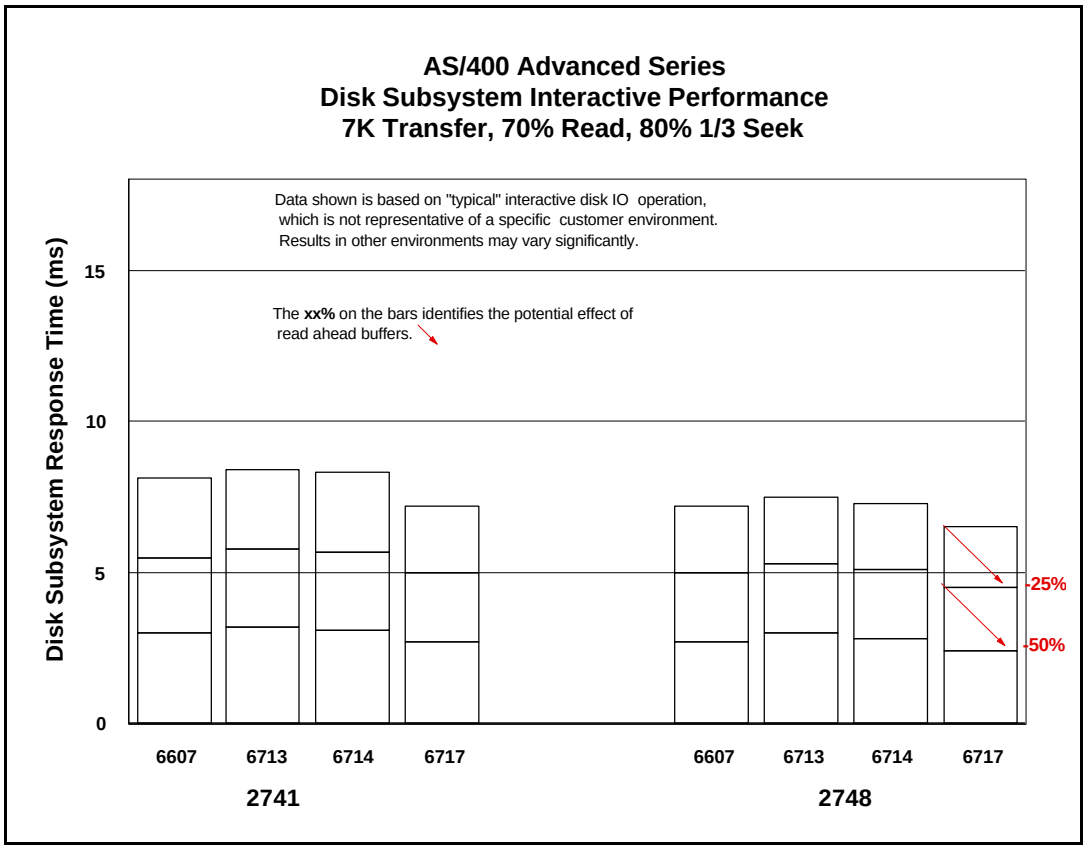



Figure 14.20. DASD Subsystem Performance - PCI RAID Controller (#2748) and 10K RPM Disks

## Conclusions / Recommendations

- The performance of previous 6607, 6713 and 6714 disks is faster with the new 2748 IOA than with the previous 2741 IOA. This performance improvement can be up to 20% faster in some cases.

- The performance on the 2741 IOA with the new 10K RPM 6717 disk is faster than with the previous 7200 RPM 6607, 6713 and 6714 disks. This performance improvement can be up to 20% faster in

some cases.

- The performance on the new 2748 IOA with the new 10K RPM 6717 disk is faster than with the previous 7200 RPM 6607, 6713 and 6714 disks on the previous 2741 IOA. This performance improvement can be up to 60% faster in some cases.

- The potential effect of device read-ahead buffers are shown for the cases of having 25% and 50% of the total disk operations already in the read ahead buffer. Depending on the data access patterns, the buffers may provide significant performance improvements.

- The above conclusions hold for batch environments also. For actual batch performance results refer to Table 14.2.

**AS/400 System Interactive Performance - PCI RAID Disk Unit Controller (#2748)**

The following graph compares the relative interactive performance of an AS/400 model 720/2064(1505) configured with 14 10K RPM (6717) DASD when running the CPW workload. The internal load source drive was ignored for this comparison chart. The curves characterize what may occur on either a 'No Protection' (also Mirrored) or RAID configuration. The graph compares the new 2748 IOA with the previous 2741 IOA.

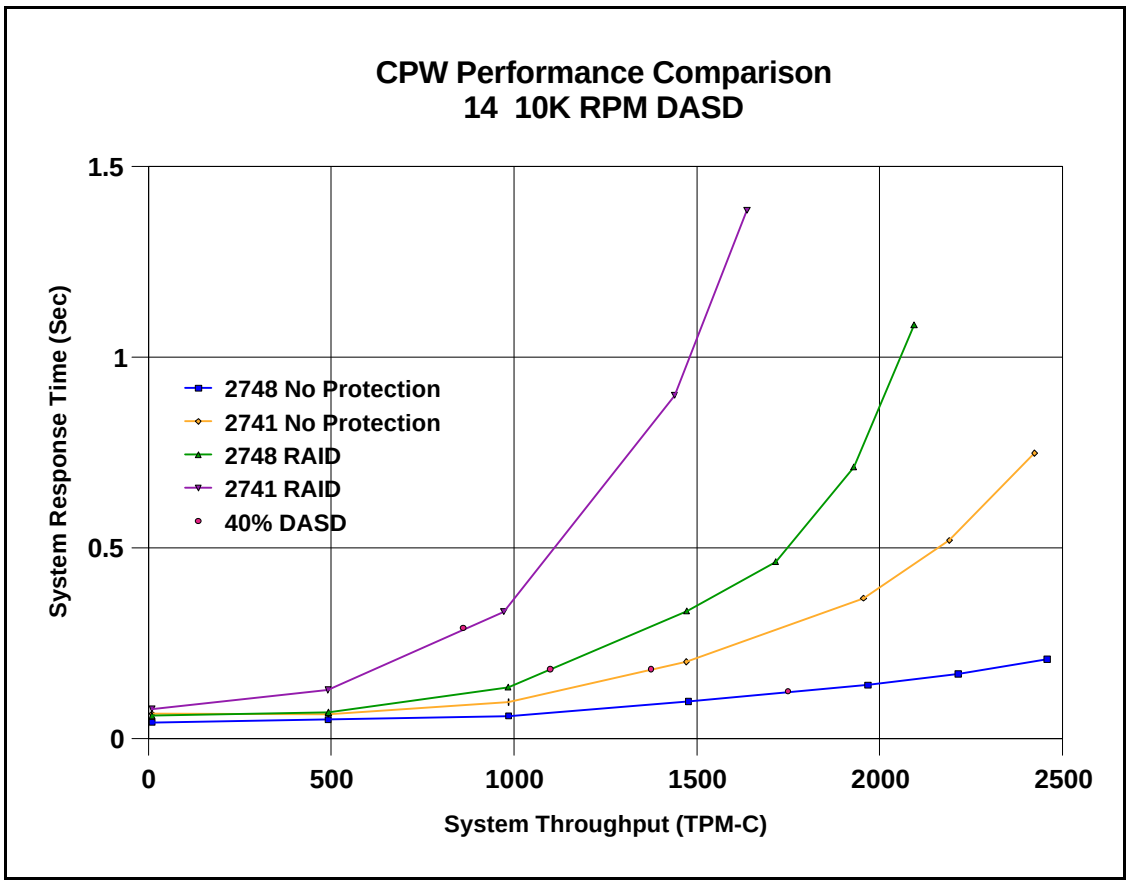



Figure 14.21. System Interactive Performance - PCI RAID Disk Unit Controller (#2748)

**Conclusions / Recommendations**

- The new 2748 PCI DASD IOA provides significantly better interactive performance than the previous 2741 DASD IOA.

- When operating with no DASD protection and at 40% DASD utilization, the system throughput improves by approximately 27% and the DASD subsystem throughput improves by about 45%.

- When operating with RAID protection and at 40% DASD utilization, the system throughput improves by approximately 28% and the DASD subsystem throughput improves by about 38%.

- The 2748 write cache (26MB) provides a significant performance advantage over the 2741 write cache (4MB).

- This graph is based on CPW workload. Other environments may vary significantly.

- The CPW benchmark's data access patterns are intentionally random, therefore, the read-ahead buffers provided only minimal benefit for CPW. Depending on your data access patterns, the DASD read ahead buffers may provide significant performance improvements.

- Similar results may occur on other AS/400 models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster AS/400 CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

**AS/400 System Interactive Performance - 10K RPM Disks**

The following graph compares the relative interactive performance of an AS/400 model 720/2064(1505) configured with 14 DASD when running the CPW workload. The internal load source drive was ignored for this comparison chart. The curves characterize what may occur on a RAID configuration. The graph compares the new 10K RPM Disks with the previous 7200 RPM Disks.

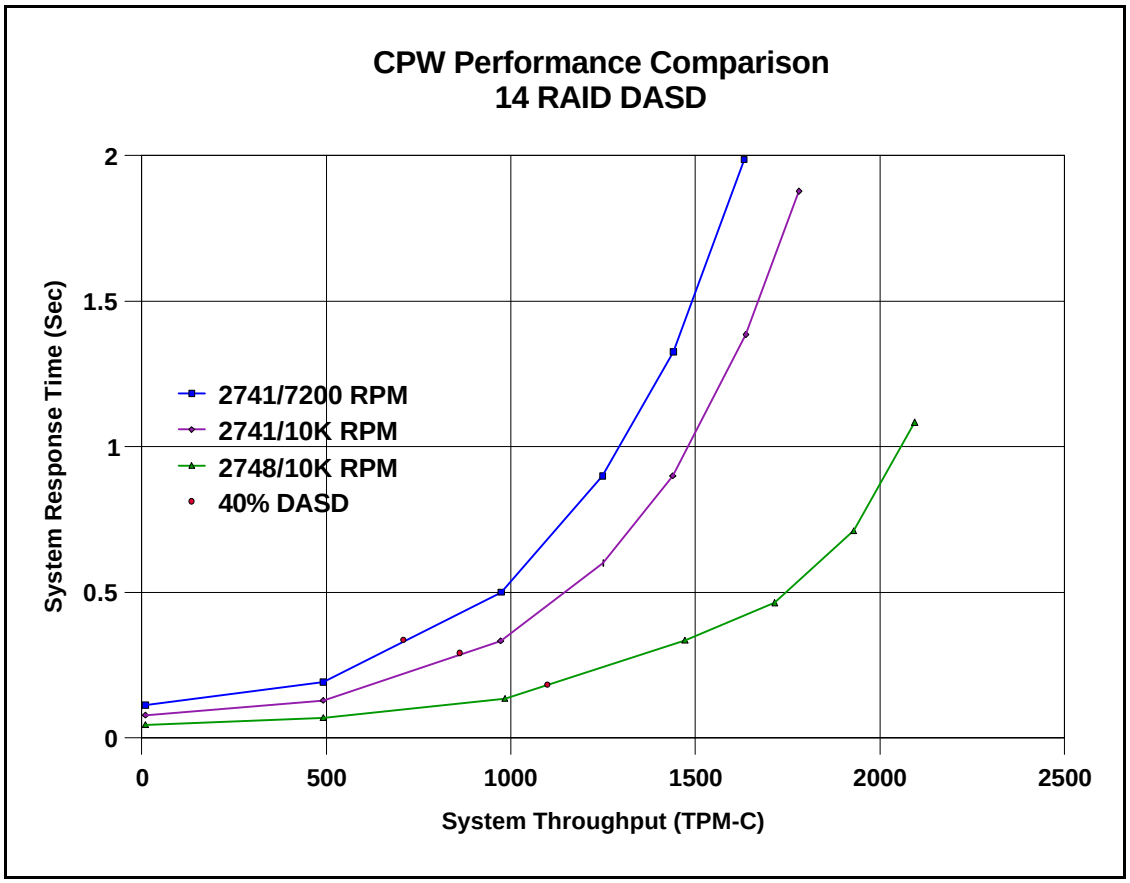



Figure 14.22. System Interactive Performance - 10K RPM Disks

**Conclusions / Recommendations**

- The new 10K RPM Disks (6717) provides significantly better interactive performance than the previous 7200 RPM Disks (6713).

- When operating with RAID protection on the 2741 IOA and at 40% DASD utilization, the system throughput improves by approximately 20% and the DASD subsystem throughput improves by about 25%.

- This graph is based on CPW workload. Other environments may vary significantly.

- The CPW benchmark's data access patterns are intentionally random, therefore, the read-ahead buffers provided only minimal benefit for CPW. Depending on your data access patterns, the DASD read ahead buffers may provide significant performance improvements.

- Similar results may occur on other AS/400 models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster AS/400 CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

## AS/400 System Interactive Performance - DASD Compression

The following graph compares the relative interactive performance of an AS/400 model 720/2064(1505) configured with the new 2748 IOA and 14 10K RPM (6717) DASD when running the CPW workload. The internal load source drive was ignored for this comparison chart. The curves characterize what may occur on this DASD subsystem configuration. The graph compares the DASD Compression/RAID with 'No Protection' and RAID Protection.

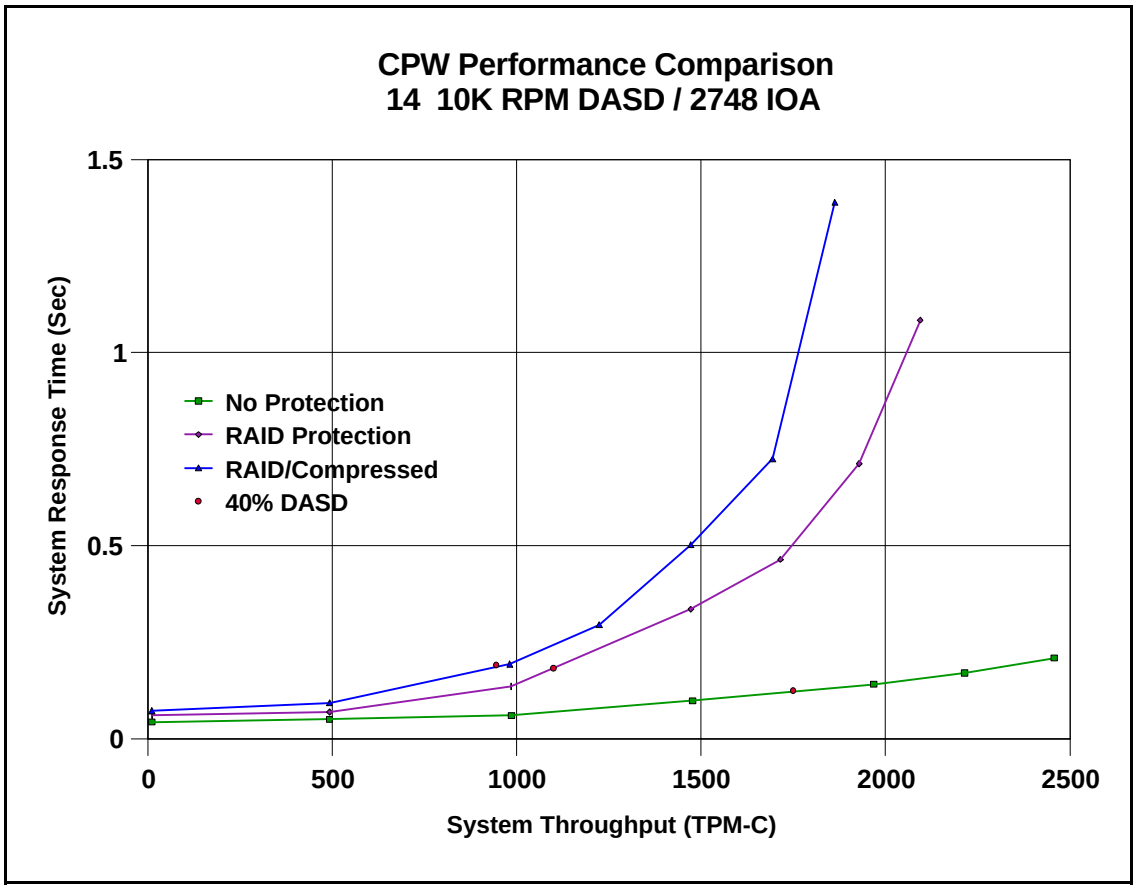



Figure 14.23. System Interactive Performance - DASD Compression

## Conclusions / Recommendations

- At lower throughputs and with equal number of disk arms, RAID/Compressed DASD has similar (0-10% degradation) system performance characteristics to RAID/Uncompressed DASD for interactive workloads.

- When operating with RAID protection and at 40% DASD utilization, the system throughput with Compression turned on is approximately 14% less than with Compression turned off.

- With the new 2748 PCI DASD IOA and 10K RPM Disks, the RAID/Compressed DASD can operate at higher throughputs than with previous 2741 DASD IOA and 7200 RPM Disks before its performance begins to deviate from RAID DASD. Just as with uncompressed DASD, the number of

disk arms must be adequate to support anticipated op rates.

- This graph is based on CPW workload. Other environments may vary significantly.

**AS/400 System Interactive Performance - Extended Adaptive Cache**

The following set of graphs compare the relative interactive performance of an AS/400 model 720/2064(1505) configured with the new 2748 IOA and 14 10K RPM (6717) DASD when running the CPW workload. The internal load source drive was ignored for these comparison charts. The curves characterize what may occur on this RAID DASD subsystem configuration when the Extended Adaptive Cache is enabled with a 1.6GB Read Cache Device. The first graph compares the Extended Adaptive Cache (EAC) on and off with Expert Cache off. The second graph compares the Extended Adaptive Cache on and off with the system level Expert Cache (EC) on and tuned for the CPW workload by using the *Usrdrn option. The third graph compares the Extended Adaptive Cache on and off with the system level Expert Cache on and NOT tuned for the CPW workload.

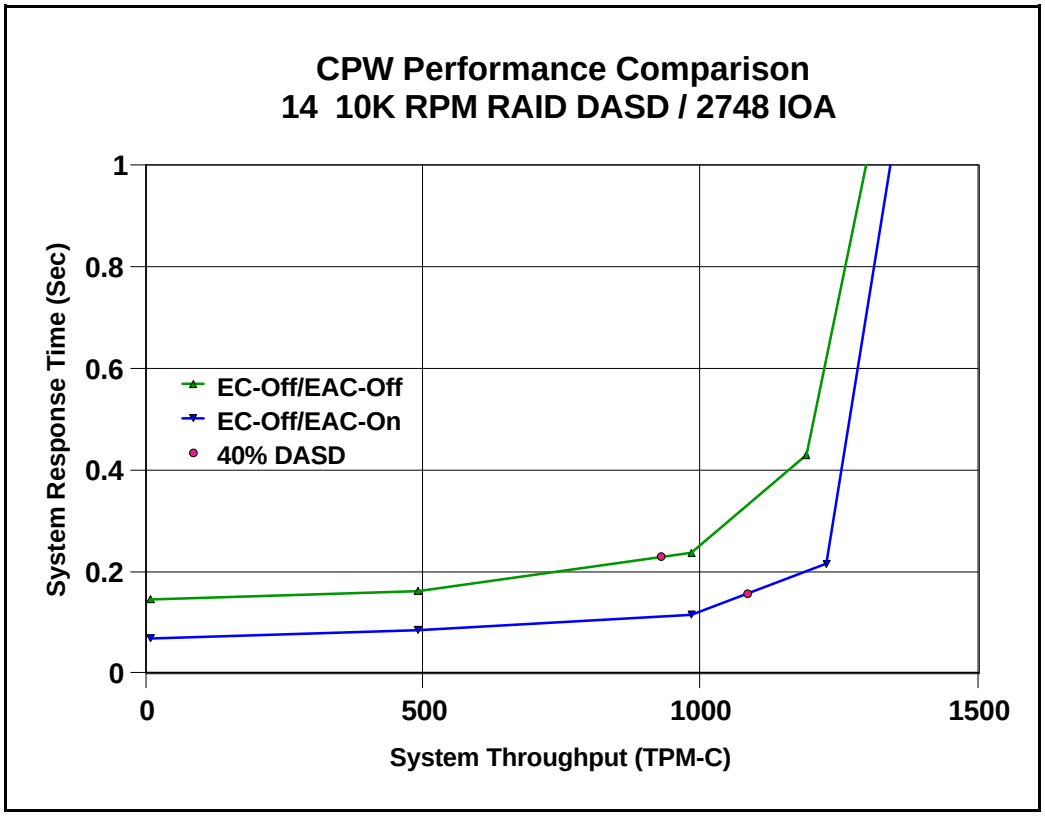



Figure 14.24. System Interactive Performance - Extended Adaptive Cache without Expert Cache

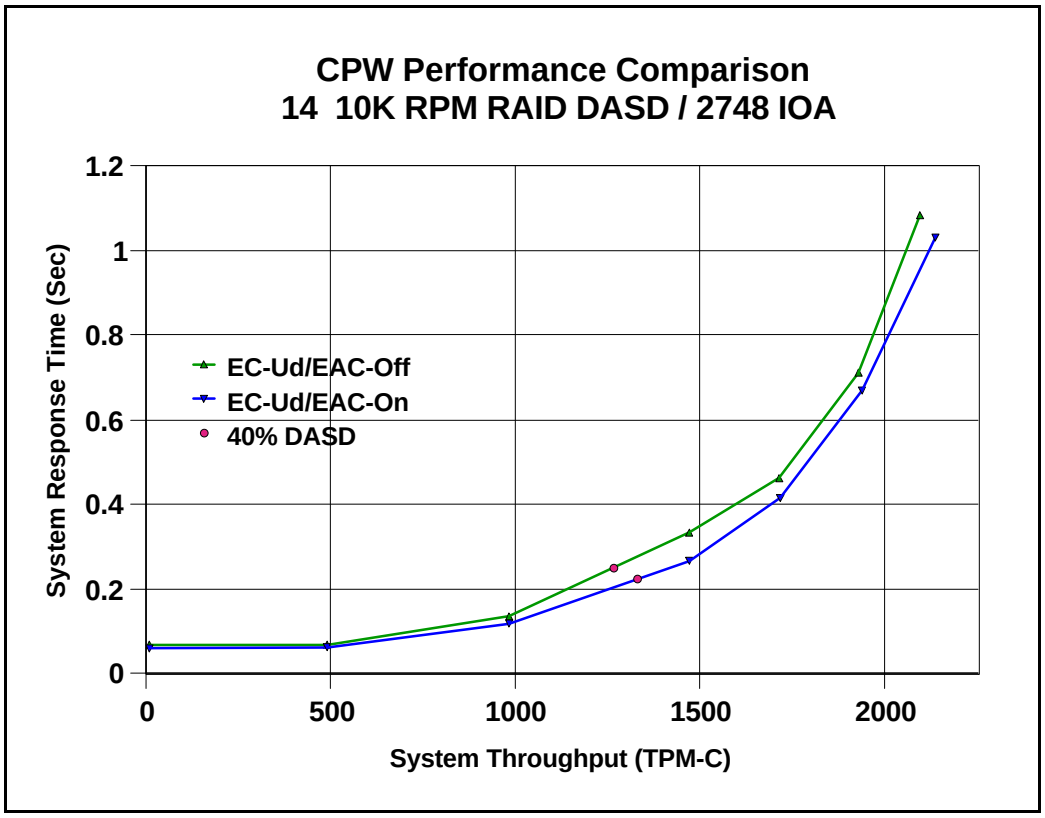



Figure 14.25. System Interactive Performance - Extended Adaptive Cache with User-Defined Expert Cache

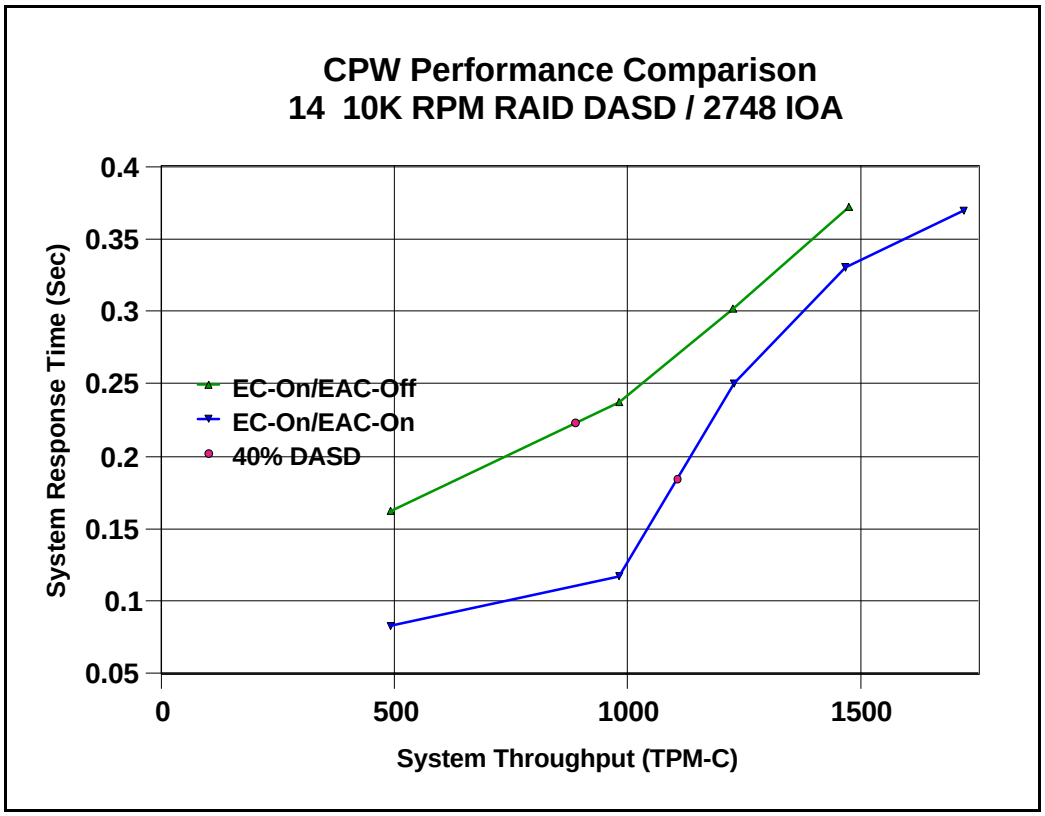



Figure 14.26. System Interactive Performance - Extended Adaptive Cache with Expert Cache

**Conclusions / Recommendations**

- The Extended Adaptive Cache (EAC-On) provides improved interactive performance whether Expert Cache is turned on or not (EC-On or EC-Off). The Extended Adaptive Cache is automatically enabled for each 2748 IOA whenever a Read Cache Device (1.6 GB solid state disk) is attached to one of its SCSI busses. Expert Cache is activated in a system memory pool by issuing an OS/400 command.

- With Expert Cache off, Extended Adaptive Cache provides significantly better interactive performance. At lower throughputs, system response time is cut in half. When operating at 40% DASD utilization, the system throughput improves by approximately 17% and the DASD subsystem throughput improves by about 26%.

- With Expert Cache on (and tuned for the CPW workload by using the *Usrdfn option), Extended Adaptive Cache provided additional interactive performance improvements over and above that given by Expert Cache. At lower throughputs, system response time was only slightly better but got faster at higher throughputs. When operating at 40% DASD utilization, the system throughput improves by approximately 6% and the DASD subsystem throughput improves by about 9%.

- With Expert Cache on (and NOT tuned for the CPW workload), Extended Adaptive Cache provided interactive performance improvements similar to that with Expert Cache off. At lower throughputs, the system response time is much faster but at higher throughputs the difference is smaller. When operating at 40% DASD utilization, the system throughput improves by approximately 20% and the DASD subsystem throughput improves by about 24%.

- This graph is based on CPW workload. Other environments may vary significantly.

**Ops/Sec/GB Guidelines for PCI RAID Controller (#2748) and 10K RPM Disks**

The metric used in determining DASD subsystem performance requirements is the number of I/O operations per second per installed GB of DASD (Ops/Sec/GB). Ops/Sec/GB is a measurement of throughput per actuator. Since DASD devices have different capacities per actuator, Ops/Sec/GB is used to normalize throughput for different capacities. An Ops/Sec/GB range has been established for each DASD type so that if the DASD subsystem performance is within the established range, the average arm percent busy will meet the guideline of not exceeding 40%.

The following bar charts show the "rule of thumb" for the Physical system Ops/Sec/GB of usable space that internal DASD subsystems can achieve with various DASD types. (To compute usable GB, we assume that the DASD subsystems have 8 disk units installed). The top of each bar is the volume of 7K transfer, 80% 1/3 seek, 30% write operations that each model can achieve when it is 40% busy.

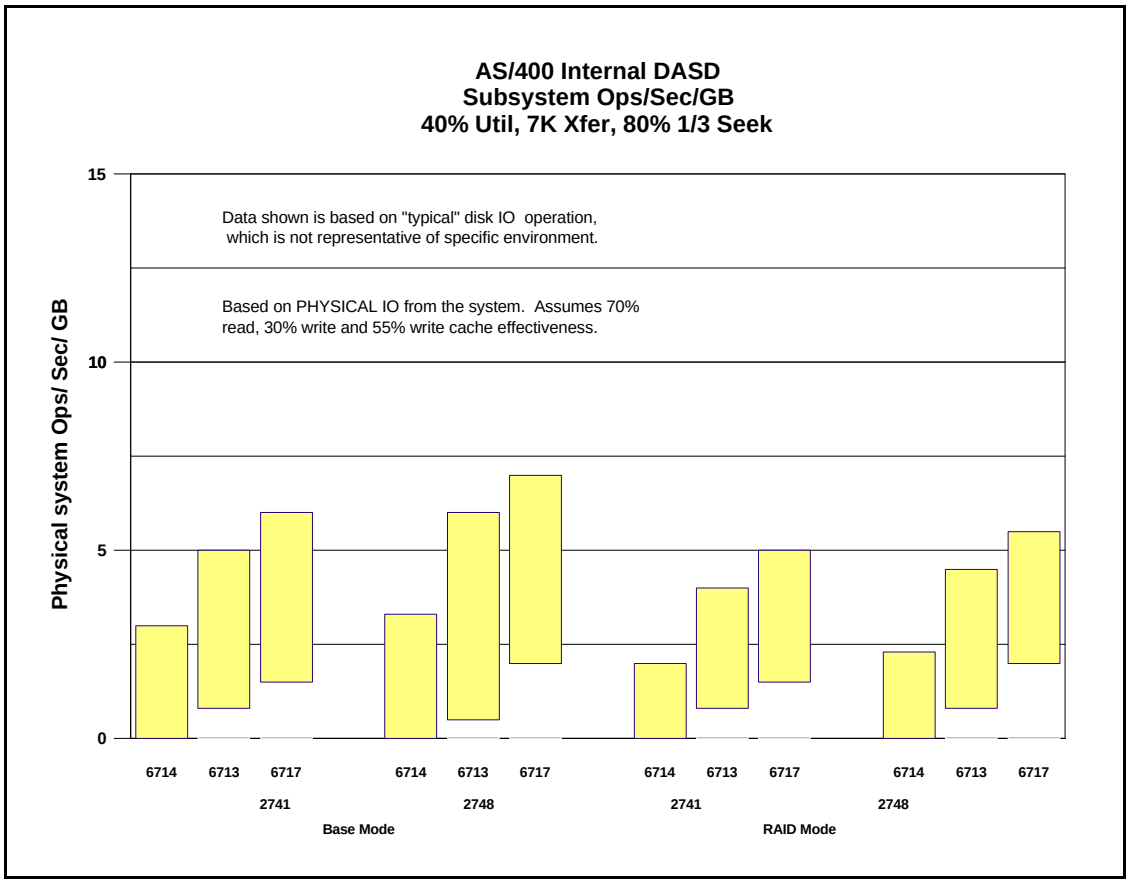



Figure 14.27. Ops/Sec/GB - PCI RAID Controller (#2748) and 10K RPM Disks

## Conclusions / Recommendations

- The new 2748 DASD subsystem provides an improved throughput operating range when compared with the 2741 DASD subsystem for both 'Base' and RAID mode.

- The new 10K RPM 9GB Disk (6717) provides an better throughput operating range when compared with the 7200 RPM 9GB Disk (6713).

## Batch Performance - PCI RAID Disk Unit Controller (#2748)

The following chart compares system performance of various batch type applications while running on an AS/400 model 720/2064(1505) configured with 14-arm user ASPs of RAID DASD. One ASP used a 2748 IOA and the other ASP used a 2741 IOA. The new 10k RPM 6717 disks were used for all of these measurements. Batch run time was measured for both of the user ASPs for 7 batch tests with the following DASD I/O characteristics :

1. Sequential read ops, 5 KB/op, OS/400 Expert Cache off

2. Sequential read ops, 60 KB/op, OS/400 Expert Cache on

3. Sequential read and write ops, 68% reads, 5 KB/op, OS/400 Expert Cache off
4. Sequential read and write ops, 17% reads, 50 KB/read op, 5 KB/write op, OS/400 Expert Cache on
5. Random read ops, 7 KB/op, OS/400 Expert Cache off
6. Random write ops, 8 KB/op, OS/400 Expert Cache off
7. Sequential read and write ops, 14% reads, 5 KB/op, OS/400 Expert Cache off

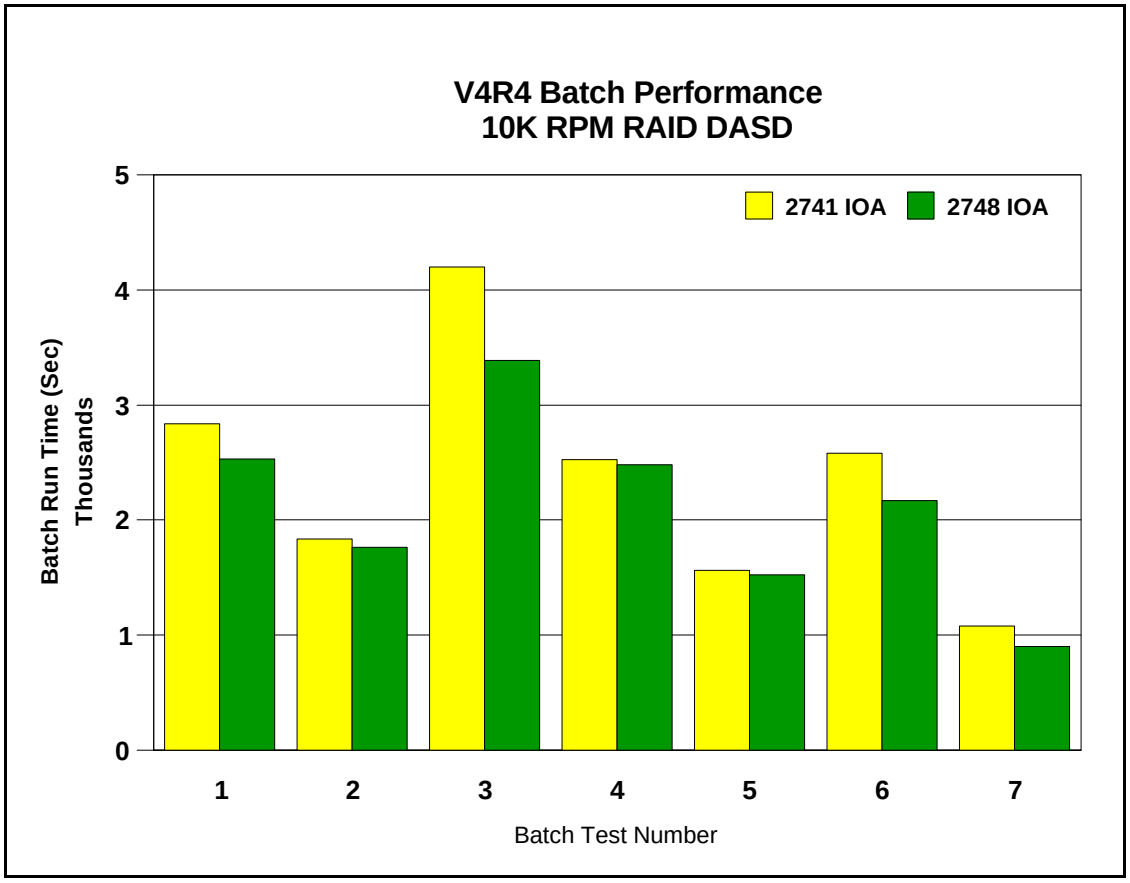



Figure 14.28. Batch Run Time Performance - PCI RAID Disk Unit Controller (#2748)

**Conclusions / Recommendations**

- The new 2748 PCI DASD IOA provides better system performance than the previous 2741 DASD IOA for all 7 of the batch tests. For tests that had write operations, the performance was significantly better. This can be attributed primarily the bigger fast write cache (26MB).
- OS/400 Expert Cache provided better batch system performance when active.

## Batch Performance - DASD Compression

The following chart compares system performance of various batch type applications while running on an AS/400 model 720/2064(1505) configured with 14-arm user ASPs of Compressed, Uncompressed, RAID/Compressed and RAID/Uncompressed DASD. The new 2748 IOA and 10k RPM 6717 disks were used for these measurements. Batch run time was measured in each of the 4 user ASPs for the 7 batch tests.

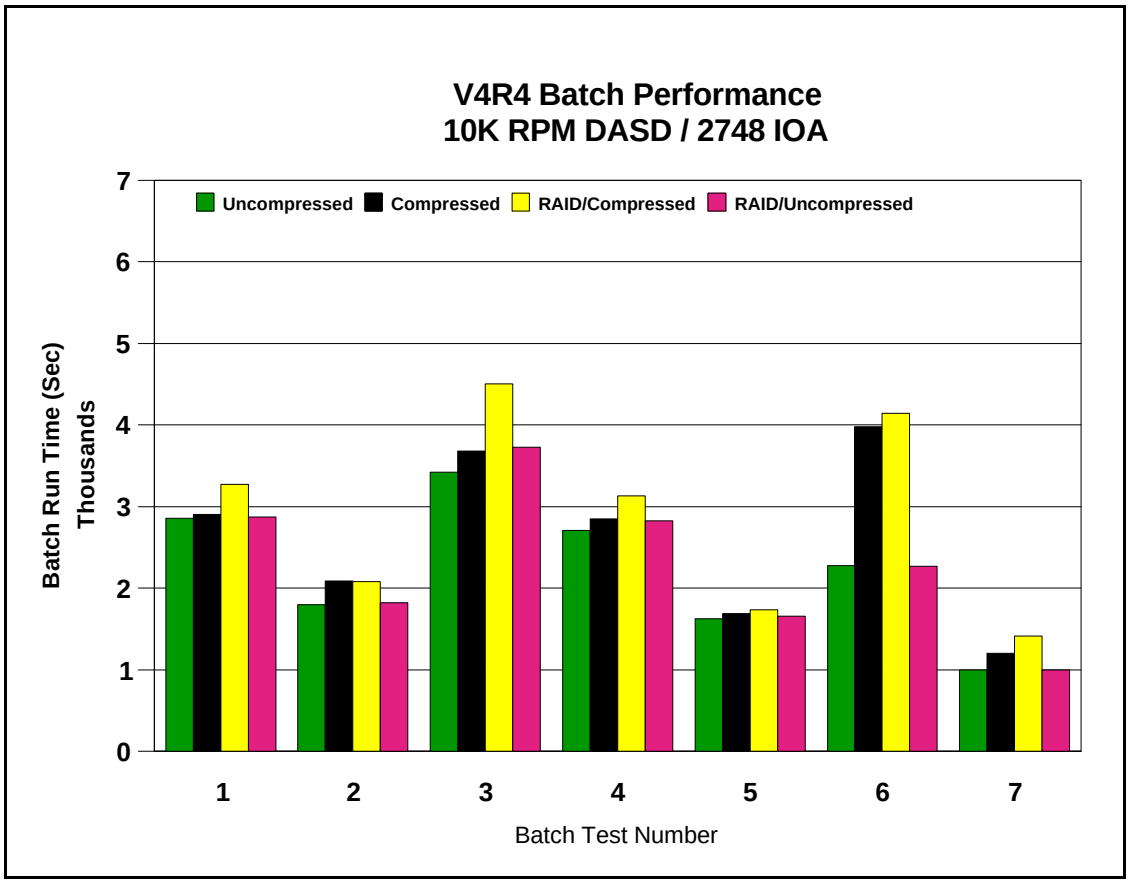



Figure 14.29. Batch Run Time Performance - DASD Compression

## Conclusions / Recommendations

- The new 2748 PCI DASD IOA and 10K RPM Disks provide better system performance for all 7 of the batch tests than was measured previously with 7200 RPM disks and the older DASD IOA (see header Batch Performance with Compressed DASD in section 14.5).

- For tests that had write operations, the performance was significantly better. This can be attributed primarily the bigger fast write cache (26MB).

- OS/400 Expert Cache provided better batch system performance when active.

## Batch Performance - Extended Adaptive Cache

The following chart compares system performance of various batch type applications while running on an AS/400 model 720/2064(1505) configured with 14-arm user ASP of RAID DASD. One ASP had the Extended Adaptive Cache enabled and the other ASP did not. The new 2748 IOA and 10k RPM 6717 disks were used for these measurements. Batch run time was measured for both of the user ASPs for the 7 batch tests.

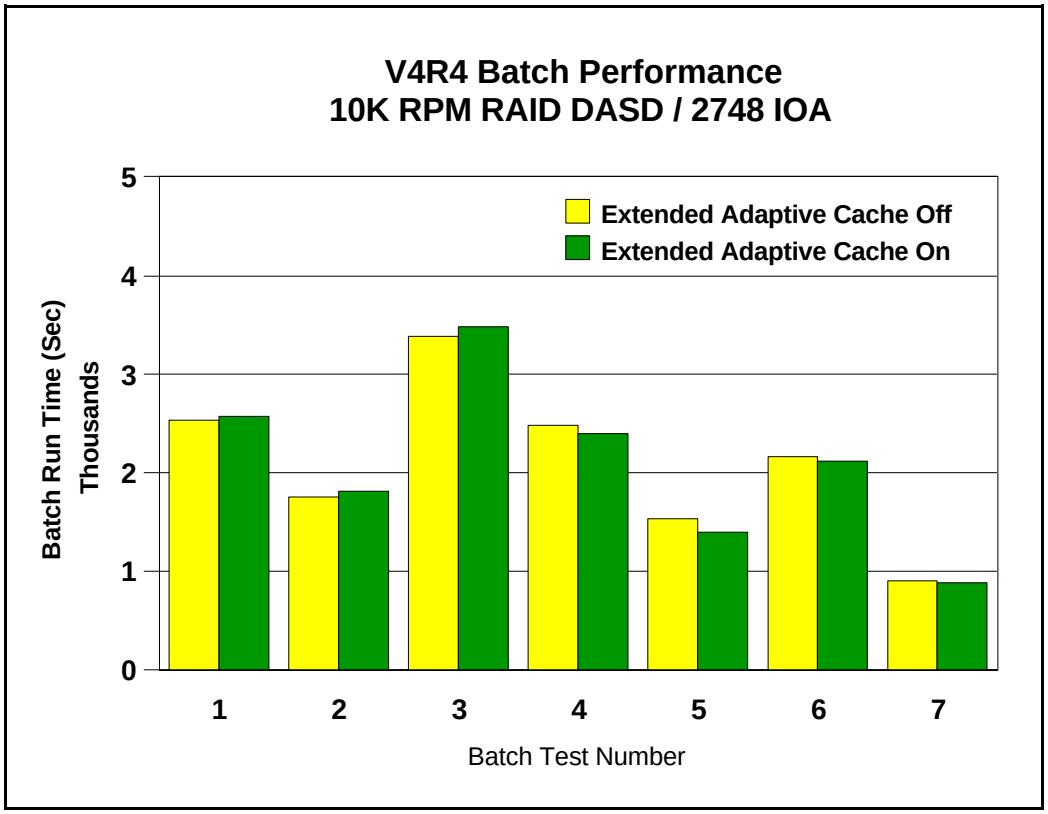



Figure 14.30. Batch Run Time Performance - Extended Adaptive Cache

## Conclusions / Recommendations

- For batch applications characterized by sequential read ops or random read ops, system performance was either only slightly less or greater with Extended Adaptive Cache enabled. This can be attributed to the fact that Extended Adaptive Cache is designed to improve performance for read-only or read/write commercial type workloads while not harming the performance of write-only, random, sequential read, or sequential write workloads. The algorithms used in Extended Adaptive Cache are NOT read-ahead based, but instead rely on locality of reference, a characteristic not exhibited by these batch applications.

- OS/400 Expert Cache provides better performance than Extended Adaptive Cache when the applications are characterized by primarily sequential read ops and should be used in these situations. The algorithms used in Expert Cache can roughly be described as read-ahead algorithms.

- OS/400 Expert Cache provided better batch system performance when active.

## 14.7 DASD Subsystem Performance Improvements for V4R5

This section discusses the DASD subsystem performance improvements that are new for the V4R5 release. These consist of the following new hardware and software offerings :

- High Speed Link (HSL) technology
- PCI RAID Disk Unit Controller (#4748/2748)
- PCI RAID Disk Unit Controller (#2763)
- 10K RPM Disks (#6718/4318)
- System Unit Expansion (#7104)
- PCI Expansion Tower (#5075)
- PCI I/O Towers (#5074/5079)

The High Speed Link technology allows I/O data to flow into and out of the system over a 1 GigaHertz bandwidth bus. This greatly reduces delays and increases overall throughput when compared with the previous SPD bus technology.

The PCI RAID Disk Unit Controller (#4748/2748) is a new DASD IOA that attaches to the PCI bus in the AS/400 models 270 and 8xx, in the PCI Expansion Tower (#5075), and in the PCI I/O Towers (#5074/5079). It provides performance improvements similar to the #2748 IOA by utilizing a Fast Write Cache of 26 MB and SCSI LVD (Low Voltage Differential Signaling) for SCSI Wide-Ultra2 (80MB) support. The 4748 IOA can connect up to 18 DASD devices and supports DASD Compression and Extended Adaptive Cache

The PCI RAID Disk Unit Controller (#2763) is a new DASD IOA that attaches to the PCI bus in the AS/400 models 270 and 820, and in the PCI Expansion Tower (#5075). It provides performance improvements over the 2740 IOA by increasing Fast Write Cache to 10 MB (from 4MB) and adding SCSI LVD (Low Voltage Differential Signaling) for SCSI Wide-Ultra2 (80MB) support on a new storage adapter. The resulting performance is similar to that of the 4748 IOA. The 2763 IOA can connect up to 12 DASD devices but does not support DASD Compression nor Extended Adaptive Cache.

The 6718/4318 is a new 10K RPM Disk (18GB) that provides faster data access than the previous 7200 RPM devices and similar data access as the 6717/4317 (9GB) devices (see figure 14.20). It can be attached only with 6532, 6533, 6751, 6754, 2726, 2740, 2741, 2748 and 2763 IOP/IOAs. It can be used as a load source and can be RAIDED and MIRRORED with its 7200 RPM counterparts.

The System Unit Expansion (#7104) when attached to the AS/400 model 270 system unit provides connectability for up to 12 additional (18 total) DASD devices. These 18 DASD devices can be connected to either 1 (#4748) or 2 (#2763 and/or #4748) PCI RAID Disk Unit Controllers in the system unit.

The PCI Expansion Tower (#5075) provides connectability of the new (#2763 or #4748) PCI RAID Disk Unit Controller and up to 6 DASD devices to a system via the High Speed Link.

The PCI I/O Tower (#5074) provides connectability of up to 3 of the new #4748 PCI RAID Disk Unit Controllers and up to 45 DASD devices to a system via the High Speed Link.

The PCI 1.8 Meter I/O Tower (#5079) provides connectability of up to 6 of the new #4748 PCI RAID Disk Unit Controllers and up to 90 DASD devices to a system via the High Speed Link.

**AS/400 System Interactive Performance - Model 820 vs. Model 720**

The following graph compares the relative interactive performance of an AS/400 model 720/2064(1505) with a new AS/400 model 820/2398(1527) when configured with a 14-arm user ASP of 10K RPM (6718) DASD running the CPW workload. The curves characterize what may occur on either a 'No Protection' (also Mirrored) or RAID configuration. The graph compares the new model 820 I/O technology with the previous model 720. On the 820 the 14 DASD were located in a #5074 PCI I/O Tower which was connected to the system by the High Speed Link. The DASD were configured under a #2748 IOA which was attached to a new and improved #2843 PCI IOP. On the 720 the 14 DASD were located in a #5065 Storage/PCI Expansion Tower which was connected to the system by the SPD bus. The DASD were configured under a #2748 IOA which was attached to a #2824 PCI IOP.

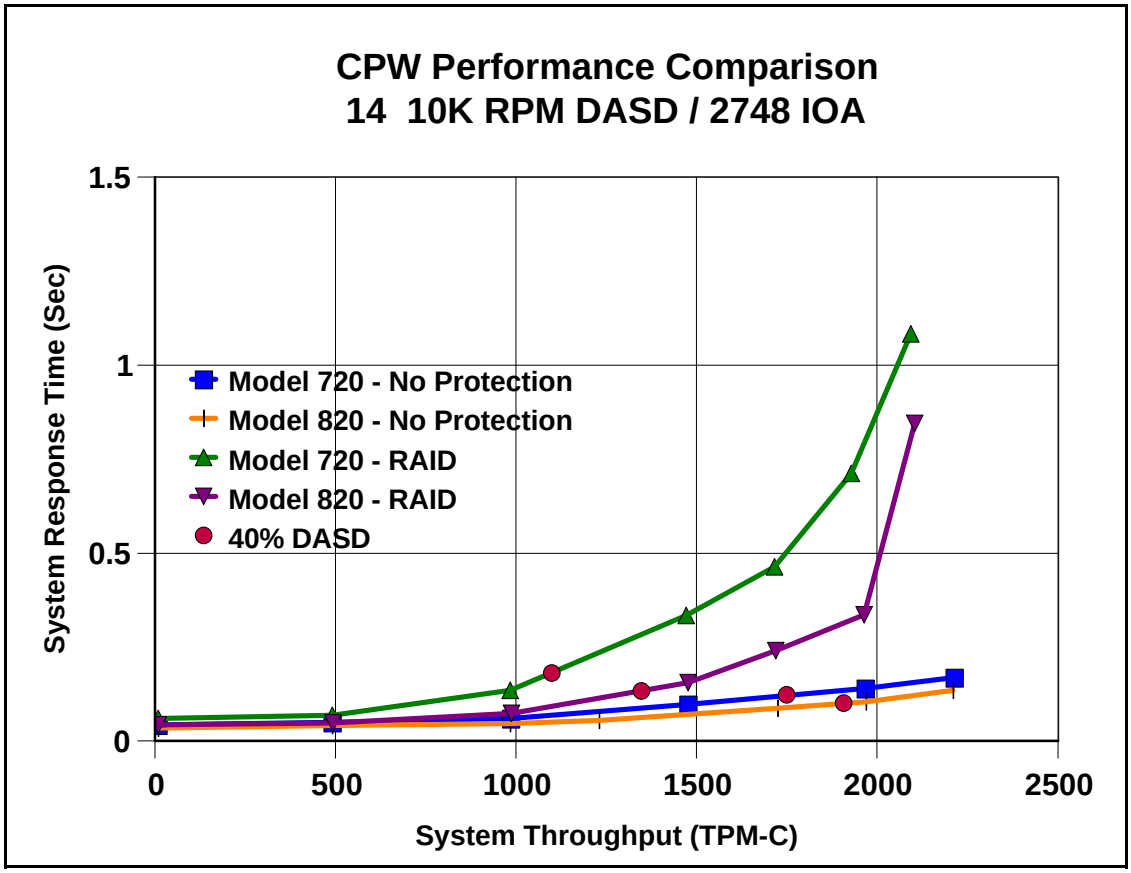



Figure 14.31. System Interactive Performance - Model 820 vs. 720

**Conclusions / Recommendations**

- The new Model 820/2398 using the same DASD IOA and 10K RPM DASD, provides significantly better interactive performance than the previous Model 720/2064. Some of the improvement can be attributed to the more powerful processors in the 820. At 40% DASD utilization, the 820 processor utilization was 5% compared to 6.8% for the 720. The rest of the performance gain is due to

improvements in the I/O and DASD subsystems such as the High Speed Link technology and more efficient PCI IOP.

- When operating with no DASD protection and at 40% DASD utilization, the system throughput improves by approximately 10% and the DASD subsystem throughput improves by about 7%.

- When operating with RAID protection and at 40% DASD utilization, the system throughput improves by approximately 20% and the DASD subsystem throughput improves by about 17%.

- This graph is based on CPW workload. Other environments may vary significantly.

- The CPW benchmark's data access patterns are intentionally random, therefore, the read-ahead buffers provided only minimal benefit for CPW. Depending on your data access patterns, the DASD read ahead buffers may provide significant performance improvements.

- Similar results may occur on other AS/400 models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster AS/400 CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

**Batch Performance - Model 820 vs. Model 720**

The following chart compares system performance of various batch type applications while running on an AS/400 model 720/2064(1505) with a new AS/400 model 820/2398(1527) when configured with a 14-arm user ASP of 10K RPM (6718) RAID DASD. The graph compares the new model 820 I/O technology with the previous model 720. On the 820 the 14 DASD were located in a #5074 PCI I/O Tower which was connected to the system by the High Speed Link. The DASD were configured under a #2748 IOA which was attached to a new and improved #2843 PCI IOP. On the 720 the 14 DASD were located in a #5065 Storage/PCI Expansion Tower which was connected to the system by the SPD bus. The DASD were configured under a #2748 IOA which was attached to a #2824 PCI IOP. Batch run time was measured for both of the user ASPs for the 7 batch tests with the following DASD I/O characteristics:

1. Sequential read ops, 5 KB/op, OS/400 Expert Cache off

2. Sequential read ops, 60 KB/op, OS/400 Expert Cache on

3. Sequential read and write ops, 68% reads, 5 KB/op, OS/400 Expert Cache off

4. Sequential read and write ops, 17% reads, 50 KB/read op, 5 KB/write op, OS/400 Expert Cache on

5. Random read ops, 7 KB/op, OS/400 Expert Cache off

6. Random write ops, 8 KB/op, OS/400 Expert Cache off

7. Sequential read and write ops, 14% reads, 5 KB/op, OS/400 Expert Cache off

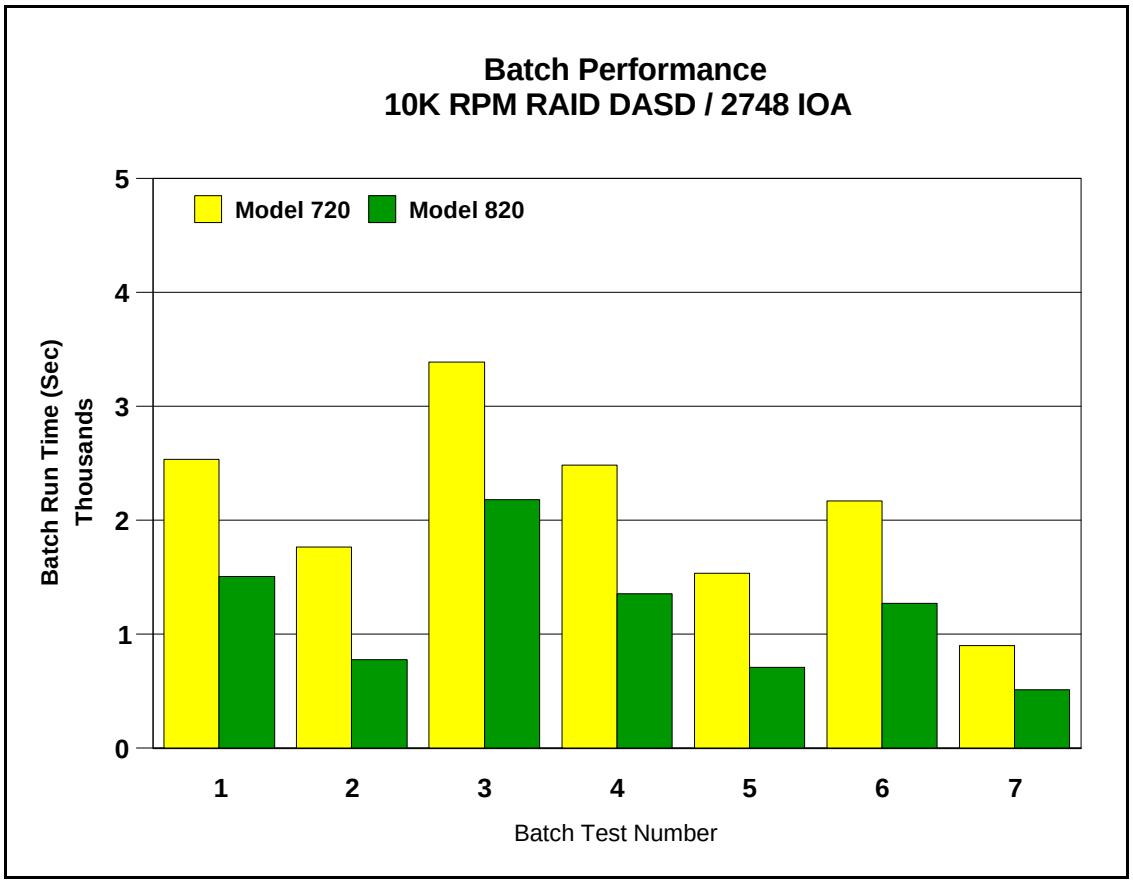



Figure 14.32. Batch Run Time Performance - Model 820 vs. 720

**Conclusions / Recommendations**

- The new Model 820/2398 using the same DASD IOA and 10K RPM DASD, provides significantly better batch performance for all of the batch tests than the previous Model 720/2064. Some of the improvement can be attributed to the more powerful processors in the 820. However, in these batch tests, the amount of CPU per I/O is relatively small. Therefore, the runtime of the test is more dependent on the DASD response time. This response time performance gain is due to improvements in the I/O and DASD subsystems such as the High Speed Link technology and more efficient PCI IOP

- The batch run time improvement varied from 56% for test 2 to 36% for test 3 and the DASD response time improvement varied form 50% to 10%.

- OS/400 Expert Cache provided better batch system performance when active.

**AS/400 System Interactive Performance - Model 270**

The following graph compares the relative interactive performance of an AS/400 model 270/2250(1516) with System Unit Expansion (#7104) feature when configured with an 18 10K RPM (6717) RAID DASD running the CPW workload. The curves characterize what may occur for the three different IOA configurations that can be used to attach these 18 DASD arms :

- A 2763 IOA attaching 6 arms and another 2763 IOA attaching the other 12 arms
- A 2763 IOA attaching 6 arms and a 4748 IOA attaching the other 12 arms
- A 4748 IOA attaching all 18 arms.

The IOAs were attached to a #2842 PCI IOP.

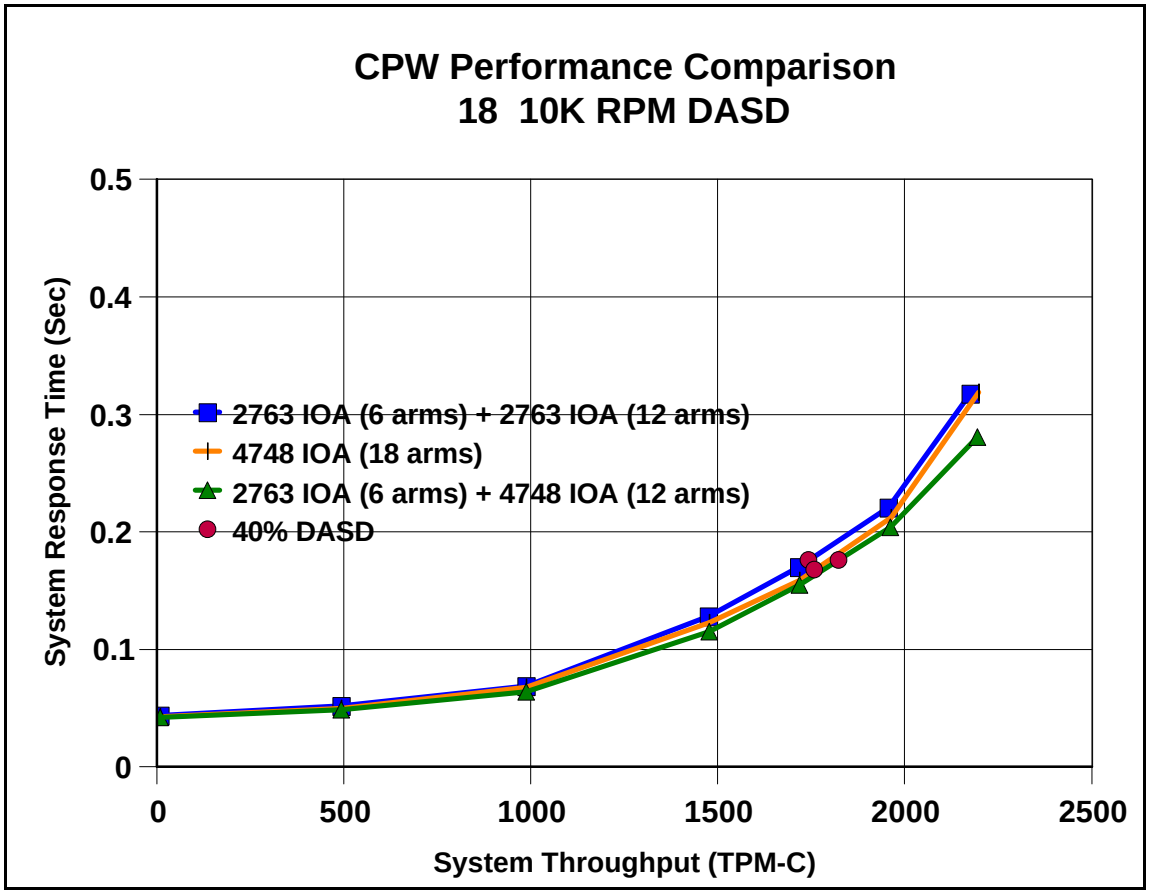



Figure 14.33. System Interactive Performance - Model 270 with System Unit Expansion feature

**Conclusions / Recommendations**

- All three DASD IOA configurations provided similar interactive performance for the low to medium throughput range. At higher throughput ranges, the 2763 IOA + 4748 IOA configuration provided the best performance.

- At 40% DASD utilization, the system throughput for 2763 IOA + 4748 IOA configuration was about 5% better than the two 2763 IOA configuration. The DASD subsystem performance was about 10% better. This performance improvement can be attributed to the better performance characteristics of the 4748 IOA when compared to the 2763 IOA.

- The 4748 write cache (26MB) provides a performance advantage over the 2763 write cache (10MB).

- If an additional PCI slot is needed, then the single 4748 IOA configuration could provide a better option.

- This graph is based on CPW workload. Other environments may vary significantly.

- Similar results may occur on other AS/400 models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster AS/400 CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

---

## 14.8 DASD Subsystem Performance Improvements for V5R1

This section discusses the DASD subsystem performance improvements that are new for the V5R1 release. These consist of the following new hardware and software offerings :

- PCI RAID Disk Unit Controller (#4778/2778)
- PCI Fibre Channel Disk Controller (#2766)

The PCI RAID Disk Unit Controller (#4778/2778) is a new DASD IOA that attaches to the PCI bus in the eServer iSeries models 270 and 8xx, in the PCI Expansion Tower (#5075), and in the PCI I/O Towers (#5074/5079). It provides performance improvements of up to 10% when compared to the #2748 IOA by utilizing a Fast Write Cache of 26MB (compression techniques effectively provide up to 104MB) and SCSI LVD (Low Voltage Differential Signaling) for SCSI Wide-Ultra2 (80MB) support. The 4778 IOA can connect up to 18 DASD devices and supports DASD Compression and Extended Adaptive Cache

The PCI Fibre Channel Disk Controller (#2766) is a new DASD IOA that attaches to the PCI bus in the eServer iSeries models 270 and 8xx, in the PCI Expansion Tower (#5075), and in the PCI I/O Towers (#5074/5079). It is used to connect the Enterprise Storage Server (ESS) /2105 DASD (code name Shark) via a high-speed Fibre Channel cable up to 10km in length. It provides performance improvements and higher band width over the older #6501 IOP. The 2843/2766 IOP/IOA combination can handle around 2660 ops/sec at maximum (100%) utilization. At 2000 ops/sec the utilization was at 70%. This is significantly better throughput than was provided by the 6501 IOP (770 ops/sec at 70%) used to attach ESS to AS/400 systems.

### eServer iSeries System Interactive Performance - PCI RAID Disk Unit Controller (#4778)

The following graph compares the relative interactive performance of an eServer iSeries model 820/2438(1527) when configured with a 14-arm user ASP of 10K RPM (6718) DASD running the CPW workload. The curves characterize what may occur on either a 'No Protection' (also Mirrored) or RAID configuration. The 14 DASD were located in a #5074 PCI I/O Tower which was connected to the system by the High Speed Link. The DASD were configured under either a #4778 IOA or #4748 IOA which was attached to a #2843 PCI IOP. The graph compares the performance of the new #4778/2778 IOA with the previous #4748/2748 IOA

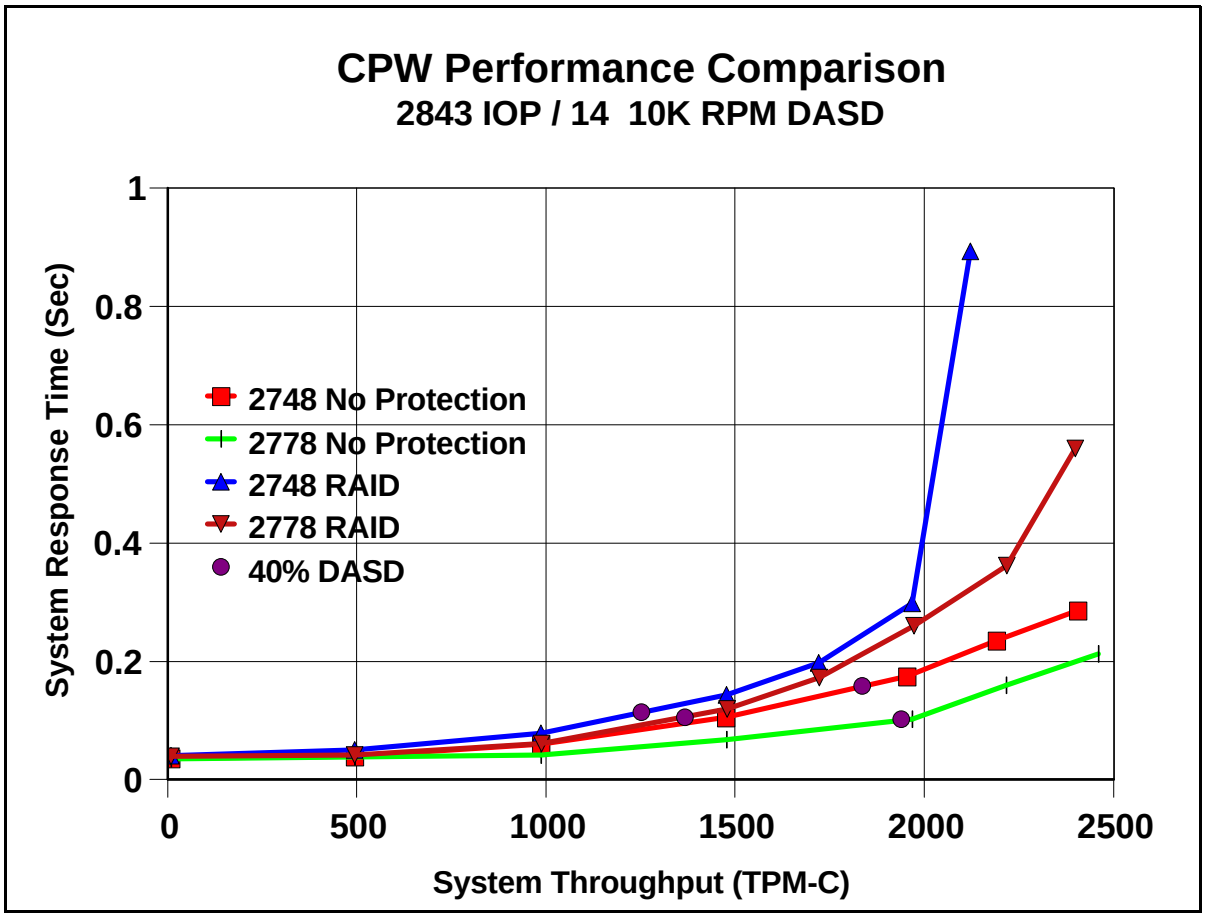



Figure 14.34. System Interactive Performance - PCI RAID Disk Unit Controller, (#4778/2778 versus #4748/2748)

**Conclusions / Recommendations**

- The new 4778/2778 PCI DASD IOA provides better (up to 10%) interactive performance than the previous 4748/2748 DASD IOA.
- When operating with no DASD protection and at 40% DASD utilization, the system throughput improves by approximately 8% and the DASD subsystem throughput improves by about 15%.
- When operating with RAID protection and at 40% DASD utilization, the system throughput improves by approximately 10% and the DASD subsystem throughput improves by about 28%.
- The 4778 compressed write cache (26MB - but compression techniques effectively provide up to 104MB) provides a performance advantage over the previous 4748 write cache (4MB). The biggest performance boost is usually seen in a heavy write or update environment.
- This graph is based on CPW workload. Other environments may vary significantly.
- The CPW benchmark's data access patterns are intentionally random, therefore, the read-ahead buffers provided only minimal benefit for CPW. Depending on your data access patterns, the DASD read

ahead buffers may provide significant performance improvements.

- Similar results may occur on other eServer iSeries models. Response time / throughput curves encounter a "knee" when a resource is used too heavily. CPU, main memory, IOP Processor and DASD are examples of resources that can cause "knees". If faster eServer iSeries CPUs are used, and other resources are unchanged, the possibility that memory or DASD will constrain the throughput increases. The BEST-1 Capacity Planner should be used to determine appropriate configurations.

## 14.9 Internal versus External DASD

There are many factors to be taken into consideration when weighing up internal vs. external DASD. The iSeries is designed for, and optimized around, complex commercial I/O through its distributed IO processor topology. This architecture uniformly distributes disk I/O and processes the work in parallel. It has therefore been expected that the natively attached disk solution, with shorter path length than external storage solutions, can exhibit better scalability and performance characteristics when compared to SAN attached storage as the workload scales.

Cross-platform disk consolidation is a major selling point for SAN storage solutions. Conceptually, this approach encourages customers to centralize their disk storage from several servers into one large server. This paper addresses performance and the key parameters that should be considered as part of any disk deployment strategy.

Benchmark tests using a simulated commercial workload environment called CPW or Commercial Processing workloads provided us with the following information:

- Fibre Channel attached I/O is considerably faster than the 6501 IOP SCSI attached external storage
- Natively attached (internal) disk generally performs better than fibre attached disk as the number of arms and therefore total capacity increases
- Expert cache is beneficial at all times for both internal and external configurations
- External read cache has a marginal but beneficial impact
- Shared disk configurations can have a significant impact on performance

For the complex commercial processing environments, we recommend that the customer start with the assumption that the disk subsystem, whether it is external or internal, be attached in a dedicated mode. Internal or natively attached disk is inherently dedicated and thus the rules for externally attached disk should follow similar configuration and performance guidelines as those used for internal, as documented in the iSeries disk arm consideration: http://www-1.ibm.com/servers/eserver/iseries/perfmgmt/diskarm.htm

For further information on iSeries internal DASD and ESS DASD comparisons refer to: http://ca-web.rchland.ibm.com/perform/perfmenu.htm and select the *IBM Enterprise Storage Server Performance on iSeries* document.

# Chapter 15. Save/Restore Performance

This chapter has been updated for V4R5/V5R1 hardware and is meant to focus on the **hardware models 270, 820, 830, 840.** For legacy system models, older device attachment cards (2729, 6501 and 6534) and the lower performing backup devices see the V4R4 performance capabilities reference. When the high speed backup devices are attached to the new 2765 and 2749 cards, the top rates can dramatically increase from what they were on the previous device attachment cards. For those systems migrating SPD towers to the new system models, save and restore rates can increase by attaching backup devices to the new 2765 and 2749 cards, and the maximum number of backup devices supported will be different than systems with only the new I/O towers. Section 15.15 for migration tower information.

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

- Hardware (such as backup device models, the number of DASD units the data is spread across)
- The backup device attachment card used.
- Type of workload (Large file, User Mix, Source File)
- The use of data compression, data compaction, and Optimum Block Size (USEOPTBLK)
- Main Storage Memory and Pool sizes

## 15.1 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 cards have different capabilities to manipulate data for the best results in their target market. The following table shows the backup devices and their rates. The rates are used later in this document to help determine possible performance. A study of some customer data showed that compaction on their data occurred at a ratio of approximately 2.7 to 1. The data in the files used for the performance workloads was created to simulate that as close as we could and compacts at no greater that 2.8:1.

*Table 15.1.1* backup device speed and compaction information

| backup Device | Rate (MB/S) | COMPACT |
|---|---|---|
| DVD-RAM | 0.75 Write #3<br>2.8 Read | 2.8 #4 |
| MLR1-S | 1.5 | 1.8 |
| MLR3 | 2.0 | 1.8 |
| SLR100 | 5.0 | 2.0 |
| 3570-C | 5.5 | 2.5 |
| 3580 SCSI #1 | 15.0 | 2.1 #2 |
| 3580 Fiber Channel | 15.0 | 2.8 |
| 3590 Ultra SCSI B model #1 | 9.0 | 2.8 |
| 3590E SCSI #1 | 14.0 | 2.4 #2 |
| 3590E Fiber Channel | 14.0 | 2.8 |

#1. The rates on these backup devices are from the 2749 card.
#2. Even though the native rate (MB/S) is a certain number, the backup devices also have a maximum throughput. We change the compaction number for the backup device to try to model what the backup device actually does.
#3. The iSeries uses the write/verify function of the backup device to assure the data integrity, so the backup device performance differs from the device specifications.
#4. Software compression is used here because the hardware doesn't support device compaction

## 15.2 Save Command Parameters that Affect Performance

### 15.2.1 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, IOP overhead, and backup device overhead. The block size does not change the IOP overhead and backup device overhead, but the number of blocks does. For example, sending 8 small blocks will result in 8 times as much IOP overhead and backup device overhead. With the larger block size, the IOP overhead and backup device overhead become less significant. 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.

### 15.2.2 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 IOP does not support data compression, the software performs the compression. This situation can require a considerable amount of processing power.

### 15.2.3 Data Compaction (COMPACT)

Data compaction is the same concept as software compression but only available at the hardware level. If you wish to use data compaction, the backup device you choose will need to support it.

## 15.3 Workloads

The following workloads were designed to help evaluate the performance of save and restore operations. Familiarization with these workloads can help in understanding differences in the save and restore rates.

**User Mix** **NUMX3GB NUMX6GB NUMX12GB** - The User mix data has been pulled into one library to create a larger sampling that can be used to evaluate concurrent and parallel save and restore operations on the newer high speed backup devices. All data was duplicated equally to create a balance between the old workload (NUMX) and the new. The old User Environment workload (NUMX) consisted of 4 libraries. The first library contains 4 source files (for a total of 1204 members) that comprise about 39 MB of space. The second library consists of 28 database files, ranging in size from 2 MB to 200 MB, which total 470 MB in size. The third library consists of 200 program objects, with an average size of about 100 KB, for a total size of 20 MB. The fourth library is 12 MB in size and consists of 2156 objects of various types. The old NUMX workload consists of about 556 MB of data.

**Source File** **NSRC1GB** - The old source file workload consisted of the 4 source files. These source files occupy about 39 MB of space and contain a total of 1204 members, and have been duplicated equally to create the new workload sizes.

**Large File** **SR4GB, SR8GB, SR16GB, SR32GB** - The large file workload is a database file. The old sampling size was 2 GB which was not a large enough sampling to evaluate parallel performance or the performance of the newer high speed backup devices so larger database files were created for evaluation purposes.

**Integrated File System** The following describes save and restore rates that a customer might see depending upon their data and its compaction capabilities. Take a system with an even mixture of client programs, such as, Lotus Notes databases and Web home pages. This example should save and restore in the range of the user mix workload described in our charts. If the data stored on the system is largely made up of database files, the save or restore rates will probably be closer to the large file type of workload rates, depending on the size and number of database files. If the data is largely made up of Web files, which tend to be numerous small HTML files such as small home pages, the save rates will be downward from user mix toward the source file workload rates.

Web objects can be large images and client databases, just as Lotus Notes database files can be numerous empty or near empty mail files. This would reverse the description above. In all situations the actual data will dictate the save and restore rates and the customer will need to know the type of data they have on their systems in order to estimate the save and restore rates.

## 15.4 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 fluctuation is dependent on the data itself.

The performance of save and restore operations is also dependent on the system configuration and the number of DASD units on which the data is stored.

Generally speaking, the large file data that was used in testing for this document was designed to compact at an approximate 2.8:1 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.

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. These factors include system size, backup device model, the amount of media that is required, and whether you are performing an attended or unattended operation.

Most of the save and restore rates listed in this document were obtained from a restricted state measurement. A restricted state measurement is performed when all subsystems are ended using the command ENDSBS SBS(*ALL), so that only the console is allowed to be signed on and running jobs. The new workloads for concurrent and parallel save and restore operations were done 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. Other subsystems such as QBATCH are required in order to run concurrent and parallel operations. All workloads were deleted before restoring them again.

## 15.5 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 15.5.1 Lower performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large File | 95% |
| User Mix | 55% |
| Source File | 25% |

**Example for a DVD-RAM:**

| DeviceSpeed | * | LossFromWorkLoad | * | Compression |
|---|---|---|---|---|
| .75 | * | .95 = (.71) | * | 2.8 = (1.995) MB/S * 3600 = 7182 MB/HR |
| .75 | * | .95 = (.71) | * | No Compression * 3600 = 2556 MB/HR |

Note: Use .95 for all workload save formulas and the other percentages for restores.

## 15.6 Medium Performing Backup Devices

The medium performing backup devices (MLR1-S, MLR3, SLR100).

*Table 15.6.1 Medium performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large File | 95% |
| User Mix | 65% |
| Source File | 25% |

**Example for SLR100:**

| DeviceSpeed | * | LossFromWorkLoad | * | Compaction |
|---|---|---|---|---|
| 5.0 | * | .95 = (4.75) | * | 2.0 = (9.5) MB/S * 3600 = 34200 MB/HR |

## 15.7 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 (3570-C, 3580, 3590E SCSI, 3590E Fiber).

*Table 15.7.1 Higher performing backup devices LossFromWorkLoadType Approximations (Save Operations)*

| Workload Type | Amount of Loss |
|---|---|
| Large File | 95% |
| User Mix | 60% |
| Source File | 12% |

**Example for 3590E:**

| DeviceSpeed | * | LossFromWorkLoad | * | Compaction |
|---|---|---|---|---|
| 14.0 | * | .95 = (13.3) | * | 2.4= (31.92) MB/S *3600 = 114912 MB/HR |
| 14.0 | * | .60 = (8.4) | * | 2.4= (20.16) MB/S *3600 = 72576 MB/HR |

**Example for 3590E Fiber:**

| DeviceSpeed | * | LossFromWorkLoad | * | Compaction |
|---|---|---|---|---|
| 14.0 | * | .95 = (13.3) | * | 2.8= (37.24) MB/S *3600 = 134064 MB/HR |

## 15.8 The Use of Multiple Backup Devices

**Concurrent Saves and Restores** - The ability to save or restore different objects from a single library to multiple backup devices or different libraries to multiple backup devices at the **same time** from **different jobs**. The workloads that were used for the testing were large file and user mix. 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 across **multiple backup devices** from the **same job**. Understand that the function was designed to help those customers, with very large 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 a better 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:

- For save and restore with a user mix and small to medium file 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 your system.

- For systems with a lot of data and a few very large files, a mixture of concurrent and parallel might be helpful. (Example: Save all of the libraries to one backup device, omitting the large files. At the same time run a parallel save of those large files to two or more additional backup devices.)

- For systems dominated by one large file 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.

- Backups where your libraries increase or decrease in size significantly throwing your concurrent saves out of balance constantly, the customer might benefit from the parallel function as the libraries 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 your system.

- Customers planning for future database 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 your BRMS/400 configuration the same but benefiting from the additional backup device. Also the same in reverse, if you lose a backup device your weekly backup doesn't have to be postponed and your BRMS/400 configuration doesn't need to change, the backup will just use the available backup devices at the time of the save.

## 15.9 Parallel and Concurrent Measurements

The 24-way system configuration we used for our testing consisted of 24 towers with 3 towers per High Speed Link (HSL). One 2749 and 45 DASD units per tower. The total of 1080 DASD units were split between 3 user Auxiliary Storage Pools (ASP's). The ASP our data was stored in had 700 DASD units. For the parallel testing the files used had 64 GB members. So the 128 GB file had 2 members the 256 GB file had 4 members, and the 512 GB file had 8 members. The 1 TB of data used was a library with 2 - 512 GB files in it. The concurrent testing used duplicate libraries with a single member 32 GB file in each. The rates obtained are extrapolated from the sample size we use and the time it takes to save that library and project that to an hour.

*Table 15.9.1 V4R5 3590E Parallel Measurements (MB/HR) (Save = S, & Restore = R)*

| # Devices | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 128 GB | S | 110000 | 215000 | 334000 | 436000 | 534000 | 604000 | | | | |
| File | R | 110000 | 195000 | 290000 | 361000 | 425000 | 448000 | | | | |
| 256 GB | S | | | | | | 652000 | 743000 | 827000 | 923000 | |
| File | R | | | | | | 436000 | 548000 | 590000 | 617000 | |
| 512 GB | S | | | | | | | | | 987000 | 1073000 |
| File | R | | | | | | | | | 627000 | 661000 |
| 1 TB | S | | | | | | | | | | 987000 |
| File | R | | | | | | | | | | 472000 |

In the table above you find that when the 512 GB file is projected over an hour we project more that 1 TB per hour but when we use a 1 TB file it winds up less than 1 TB per hour. This is because when the parallel run actually spans multiple backup devices, the backup devices have to wait for the save job to let them eject their media and load a new one. Since this comes from a single save job only one backup device can be switching media at a time. The difference gets more and more noticeable as the number of backup devices used increases. The saves to all the backup devices start within seconds of each other, so that all of the media in the backup devices will fill at about the same time. When the backup devices are ready to switch media, they all wind up sitting as the first backup device switches media and then starts the save again. Then the second backup device starts switching media and the others wait until that one is done, and so on. For the rest of your save the backup devices are at different stages of full, so the switching shouldn't drag the save time down from there on. The second affect of the media switching like this, is that the media is not evenly filled when the save job is complete. This can create a problem for the customer when planning the number of media units to load in each backup device.

The measurements using 24 backup devices were done to show that the new system HSLs are prepared for the future. On the older models systems the maximum through put for a save operation was around 350 GB/HR and this shows we are able to run 24 concurrent saves totaling 2.7 TB/HR (Averaging 112 GB/HR/Device). We believe the 840 model limit is somewhere around 4 TB/HR but can't project what would happen with DASD and CPU at that point. The test also showed that a parallel save held flow pretty well up to the 24 backup devices (Averaging 91 GB/HR/Device). This test was conducted on our 840 24 way system with 24 towers and 96 GB of mainstore memory.

*Table 15.9.2 V4R5 3590E 24 backup Device Measurements (MB/HR)*

| Concurrent operations | 24 backup devices |
|---|---|
| Save | 2700000 |
| Restore | 700000 |
| Parallel Operation | |
| Save | 2220000 |
| Restore | 525000 |

## 15.10 Maximum Number of Backup Devices on a System

Identifying the maximum number of backup devices for the different system models can be difficult with all the factors to consider. If the DASD, memory and CPU are at their maximum and we are saving a large file workload then the system might be able to achieve the maximum save limit. For a 270 model system the limiting factor would be the number of DASD arms which would probably limit the system to one high speed backup device or about 100 GB/HR for large file data. A single processor could feed two backup devices with large file data but the number of DASD wouldn't support the operations. If memory and DASD were at their maximum on the 820, 830, or 840 the limiting factor would be the processors. For large file data, two backup devices per processor can be possible.

## 15.11 How the Number of Processors Affects Performance

With the large file workload we have been able 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.

## 15.12 DASD and Backup Devices Sharing a Tower

The new system architecture does not require DASD and backup devices be kept separated. In the testing in the IBM Rochester Lab we attached one backup device to each tower and all towers had 45 DASD units in them. You aren't limited to putting one backup device in a tower, but in order to supply multiple backup devices the system needs enough DASD units to feed the backup devices. We advocate spreading your backup devices amongst the towers available.

## 15.13 How Memory Pool Size Affects Performance

These measurements were on an 840 12-Way. 45 DASD units, DASD units raid protected. This is an attempt to show that memory in a pool can effect the save or restore operation. If the system is restricted the save and restore operations assume all memory belongs to that job and adjusts accordingly.

*Table 15.13.1 Limit Number of Jobs/Memory Measurements on a 3590E (MB/HR)*

| | 16 GB File Save | 16 GB File Restore | 6GB User Mix Save | User Mix Restore |
|---|---|---|---|---|
| Memory = # Jobs<br>Pool Size Max. Act<br>10000.00 10000 | 54000 | 56000 | 40000 | 40000 |
| Memory = 2X # Jobs<br>Pool Size Max. Act<br>10000.00 5000 | 84000 | 56000 | 58000 | 40000 |
| Memory = 3X # Jobs<br>Pool Size Max. Act<br>10000.00 3300 | 94000 | 56000 | 62000 | 40000 |

Note: The system value QPFRADJ can be set on allowing the system to tune itself so that the customer doesn't have to worry about the memory pools. The tuner will need time to adjust to the new work flow so it is more effective on a longer save.

## 15.14 How the number of DASD Units affects Performance

With our old workloads (NUMX and 2 GB) the sampling size is too small and the overhead it takes to save the small amount of data doesn't allow us to extrapolate what will really happen if larger amounts of that data are saved or restored over time. So for the faster backup devices we have created larger samples. Some of these samplings still only run for minutes on the newer backup devices, but were trying to use a large enough sampling that would actually run for an hour. The following charts show the affects the number of DASD units have on the save and restore operations. These can help our customers identify some of the things that are common to their systems and help determine what solutions might be right for their situation. The following measurements were on an 840 12-way system. All DASD units were raid protected.

*Table 15.14.1 Limit DASD Units Measurements on a 3590E (MB/HR)*

| | 16 GB File Save | 16 GB File Restore | 6GB User Mix Save | 6GB UserMix Restore |
|---|---|---|---|---|
| 15 DASD Units | 98000 | 32000 | 56000 | 23000 |
| 30 DASD Units | 98000 | 40000 | 61000 | 29000 |
| 45 DASD Units | 98000 | 59000 | 63000 | 40000 |
| 60 DASD Units | 112000 | 100000 | 75000 | 45000 |
| 75 DASD Units | 112000 | 112000 | 81000 | 45000 |
| Note: The following concurrent numbers are used to show that the number of DASD Units effect the operations. Keep in mind that these numbers are gathered using a large file workload and that a user mix workload needs a larger number of DASD arms to feed it as in the examples above. | | | | |
| 16 GB File | Concurrent Save 75 DASD Units | Concurrent Restore 75 DASD Units | Concurrent Save 90 DASD Units | Concurrent Restore 90 DASD Units |
| 1 3590E's | 110 | 115 | 110 | 115 |
| 2 3590E's | 218 | 138 | 218 | 160 |
| 3 3590E's | 315 | 145 | 315 | 165 |

## 15.15 Migrations towers attaching SPD

For those customers choosing to migrate their existing SPD busses and attached DASD towers, there are some conceptual limits that are different than those systems using purely new hardware. For the most part a system with migration towers just needs to satisfy all of the other rules listed about memory, number of DASD, number of processors, and the backup devices must be attached to the new 2765 and 2749 cards. The links that attach the migration towers will peak before the new links to the new towers, but for most customers with a few high speed backup devices it won't have a noticeable affect. There will be a limit to the number of backup devices a customer can make use of but it will be unique to the mix of SPD and HSL towers attached and the data being saved.

## 15.16 Slower Save After an IPL

In some cases customers are experiencing a longer save time on the first save after an IPL. The cause for this seems to originate in the following areas.
1. Objects go through additional checking when they are first touched after an IPL and the subsequent touches of the object can be much faster. This first touch of an object is an initialization of the object so that there is a temporary working space for that object to be used.
2. Portions of objects are paged into memory allowing subsequent accesses of the objects without accessing the disk.

There are a few actions we can recommend if this condition significantly affects you.
1. Limit your IPLs. If you know you must IPL for PTFs or some other reason understand that the next save will be longer and plan accordingly.
2. Carefully consider the sequencing of your IPLs and saves. Since a save done after an IPL will be slower, if you have a short window of time to get them both completed, then consider doing the save before the IPL. On the other hand, if you have a sufficiently large window of time, you may want to do the IPL first, so that the save that follows will do first touch initialization/paging-in of the objects and thus help speed up subsequent operations.
3. Create a simple CL program and attach it to the system start up program to be started after an IPL. This would be a low priority batch program that would go out and touch objects on the system. Then if you have enough time for this to complete between the IPL and the first save the DSPFD will take care of the first touch of the database objects.
Step 1. CRTDUPOBJ OBJ(QAFDMBRL) FROMLIB(QSYS) OBJTYPE(*FILE) TOLIB(USERLIB) NEWOBJ(FDMBRL)
Step 2. CHGPF FILE(USERLIB/FDMBRL) SIZE(*NOMAX)
Step 3. Command to submit
SBMJOB CMD(DSPFD FILE(*ALL/*ALL) TYPE(*MBRLIST) OUTPUT(*OUTFILE) FILEATR(*PF *LF) OUTFILE(USERLIB/FDMBRL) OUTMBR(FDMBRL *REPLACE) ) JOB(DSPFDMBRL) JOBQ(QSYSNOMAX)

Manual in the associate info APAR II12893

## 15.17 V5R1 Rates

*Table 15.17.1 - V5R1 Measurements on an 270 1-way system 6 RAID protected DASD Units (MB/HR)*

| | NSRC1GB | NUMX3GB | NUMX12GB | 4GB | 8GB | 16GB |
|---|---|---|---|---|---|---|
| DVD RAM DTACPR *YES | | | | | | |
| SAVE | 6500 | 6500 | | 6500 | 6500 | |
| RESTORE | 3100 | 9500 | | 12000 | 13000 | |
| DVD RAM DTACPR *NO or *DEV | | | | | | |
| SAVE | 2500 | 2500 | | 2600 | | |
| RESTORE | 2600 | 6600 | | 9400 | | |
| SLR100 | | | | | | |
| SAVE | 7300 | 21000 | 22000 | 32000 | 33000 | 34000 |
| RESTORE | 2300 | 10000 | 10000 | 19000 | 19000 | 19000 |

*Table 15.17.3 - V5R1 Measurements on an 840 24-way system <200 RAID protected DASD Units (MB/HR)*

| | NSRC1GB | NUMX3GB | NUMX12GB | 4GB | 8GB | 16GB |
|---|---|---|---|---|---|---|
| DVD RAM DTACPR *YES | | | | | | |
| SAVE | 7000 | 7000 | | 7000 | 7000 | |
| RESTORE | 6400 | 15000 | | 19000 | 19000 | |
| DVD RAM DTACPR *NO or *DEV | | | | | | |
| SAVE | 2500 | 2500 | | 2600 | | |
| RESTORE | 5200 | 8800 | | 9500 | | |
| SLR100 | | | | | | |
| SAVE | 10000 | 23000 | 23000 | 33000 | 34000 | 34000 |
| RESTORE | 7900 | 23000 | 23000 | 33000 | 34000 | 34000 |
| 3580 SCSI | | | | | | |
| SAVE | 15000 | 61000 | 68000 | 90000 | 99000 | 104000 |
| RESTORE | 8500 | 43000 | 43000 | 92000 | 100000 | 104000 |
| 3580 FIBER | | | | | | |
| SAVE | 16000 | 70000 | 78000 | 120000 | 131000 | 139000 |
| RESTORE | 8500 | 40000 | 40000 | 120000 | 130000 | 139000 |
| 3590E SCSI | | | | | | |
| SAVE | 14000 | 71000 | 73000 | 108000 | 110000 | 114000 |
| RESTORE | 8800 | 46000 | 46000 | 111000 | 112000 | 115000 |
| 3590E FIBER | | | | | | |
| SAVE | 18000 | 78000 | 78000 | 130000 | 130000 | 136000 |
| RESTORE | 9000 | 50000 | 50000 | 125000 | 125000 | 132000 |

## 15.18 V4R5 Rates

*Table 15.18.1 - V4R5 Measurements on an 270 1-way system 6 RAID protected DASD Units (MB/HR)*

| | | NUMX | NSRC | DLO | 2 GB File |
|---|---|---|---|---|---|
| MLR1 | Save | 7500 | 2900 | 4500 | 9300 |
| | Restore | 7400 | 3000 | 4100 | 9500 |
| 3590B | Save | 25000 | 5000 | 20000 | 75000 |
| | Restore | 11000 | 3000 | 5200 | 22000 |
| 3590E | Save | 25000 | 5000 | 26000 | 79000 |
| | Restore | 11000 | 3000 | 5200 | 22000 |

*Table 15.18.2 - V4R5 Measurements on an 820 1-way system 12 RAID protected DASD Units (MB/HR)*

| | | NUMX | NSRC | DLO | 2 GB File |
|---|---|---|---|---|---|
| MLR3 | Save | 9000 | 2800 | 5600 | 12000 |
| | Restore | 9000 | 2800 | 4700 | 12000 |
| 3590B | Save | 26000 | 6500 | 20000 | 75000 |
| | Restore | 17000 | 4000 | 7300 | 33000 |
| 3590E | Save | 36000 | 6500 | 29000 | 95000 |
| | Restore | 17000 | 4000 | 7300 | 33000 |

*Table 15.18.3 - V4R5 Measurements on an 820 1-way system 57 RAID protected DASD Units (MB/HR)*

| | | NUMX6G | 16GB Large File |
|---|---|---|---|
| 3590E | Save | 71000 | 114000 |
| | Restore | 42000 | 114000 |

*Table 15.18.4 - V4R5 Measurements on an 820 4-way system 45 RAID protected DASD Units (MB/HR)*

| | | NSRC | NUMIX6G | 16 GB File |
|---|---|---|---|---|
| MLR3 | Save | 3000 | 12000 | 12000 |
| | Restore | 3000 | 10000 | 12000 |
| 3590B | Save | 10000 | 69000 | 85000 |
| | Restore | 3000 | 42000 | 84000 |
| 3590E | Save | 10000 | 80000 | 112000 |
| | Restore | 7000 | 42000 | 84000 |

*Table 15.18.5 - V4R5 3590E Measurements on an 840 24-way system <200 RAID protected DASD Units (MB/HR)*

| | | NSRC | NUMIX6G | 4 GB File | 16 GB File |
|---|---|---|---|---|---|
| 3590 | Save | 7500 | 85000 | 99500 | 115000 |
| | Restore | 3000 | 40000 | 101000 | 115000 |

## 15.19 What's New and Tips on Performance

### What's New

1. 2765 Fiber card for attaching the 3590E and 3580 Fiber model tape drives.

2. 3590E Fiber Channel tape drive, 3580 Fiber Channel tape Drive, DVD RAM drive.

3. In general if the high speed backup devices (3580, 3590E) are attached to the new 2749 cards the performance increase for that save operation could be from 25% to 50% over the same backup device attached to a 6534 card. However there will be those customers who have data that is already saving as fast as the data allows and there will be no improvement in save times.

### TIPS

1. Backup devices are effected by the media type. For most backup devices the right media and density can greatly effect the capacity and speed of your save or restore operation. **USE THE RIGHT MEDIA FOR YOUR BACKUP DEVICE.**

2. Using the default setting for the USEOPTBLK parameter of *YES on save commands can significantly improve performance on newer backup devices. This is especially true where the system's CPU is subjected to a heavy workload.

3. A Backup and Recovery management system such as BRMS/400 is recommended to keep track of the data and make the most of multiple backup devices.

4. Save-While-Active performance improvements to reduce checkpoint processing time (one test with a sample SAP library showed that checkpoint processing that previously took 19:36 minutes now took only 1:36 minutes).

5. Performance improvement when restoring (or creating, etc.) large numbers of IFS files -- the more files (e.g., million+), the bigger the improvement. PTF SF65133 is available for V4R5 and SF65355 for V4R4.

# Chapter 16 IPL Performance

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

The primary focus of this section is to present data that compares V4R5 IPL times and V5R1 IPL times using two hardware configurations. The data for both a normal and abnormal IPL is broken down into phases, making it easier to see the detail.

**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.

## 16.1 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 is a simple description of the IPL tests performed and documented here.

## 16.2 IPL Benchmark Description

Normal IPL

- Power On IPL (cold start after system was shut off)
- For a normal IPL, benchmark time is measured from power-on to console sign-on screen

Abnormal IPL

- System abnormally terminated causing recovery processing to be done during the IPL. The amount of processing is determined by the activity and reason the system terminated.
- 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). System key switch 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 the time the console sign on screen appears.
- 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.

### 16.2.1 Large System Benchmark Information

**Hardware Configuration**
840-23FE(24-way) with 128 GB Mainstore
DASD / 1080 units
100 Dasd units in ASP 1 mirrored, 800 Dasd units in ASP 2 RAID protected
3 ASP's defined
Mainstore dump was to ASP 2

**Software Configuration**
90,000 spool files (30,000 completed jobs with 3 spool files each)
1000 jobs waiting on job queues (inactive)
11000 active jobs in system during mainstore dump
200 remote printers
6000 user profiles
3000 libraries

**Database:**
- 25 libraries with 2600 physical files and 452 logical files
- 2 libraries with 10,000 physical files and 200 logical files
- This system was tested with 4 TB of database files unrelated to this test, but this filled the dasd units to 50% which causes a long directory recovery. See section 16.4 for information.

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

### 16.2.2 Small System Benchmark Information

**Hardware Configuration**
270-22A4 with 4 GB Mainstore
DASD / 6 arms, 51 GB,
RAID Protected

**Software Configuration**
2,000 spool files (2,000 completed jobs with 1 spool file per job)
350 jobs in job queues (inactive)
500 active jobs in system during Mainstore dump
100 user profiles
200 libraries

**Database:**
- 1 library with 100 physical files and 20 logical files
- 1 library with 50 physical files and 10 logical file.

## 16.3 IPL Performance Measurements

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

Measurement units are in minutes and seconds

*Table 16.3.1 Normal IPL Benchmark Summary - Power-On (Cold Start)*

| | Large System | | Small System | |
|---|---|---|---|---|
| | **V4R5 24 Way 840-23FE** | **V5R1 840-23FE** | **V4R5 270-22A4** | **V5R1 270-22A4** |
| **Hardware** | 7:09 | 7:20 | 3:50 | 4:30 |
| **SLIC** | 8:34 | 9:12 | 1:57 | 2:30 |
| **OS/400** | 5:50 | 4:50 | 3:22 | 1:45 |
| **Total** | 21:34 | 21:22 | 9:11 | 8:45 |

Generally, the hardware phase is composed of C1xx xxxx and C3xx xxxx SRCs, SLIC is composed of C600 xxxx SRCs, and OS/400 is composed of C900 xxxx SRCs plus time to console sign-on

Measurement units are in hours, minutes and seconds.

*Table 16.3.2 Abnormal IPL Benchmark Summary*

| | Large System | | Small System | |
|---|---|---|---|---|
| | **V4R5 24 Way 840-23FE 96GB MS** | **V5R1 24 Way 840-23FE 128GB MS** | **V4R5 270-22A4** | **V5R1 270-22A4** |
| **Processor MSD** | 28:02 | 24:42 | 5:10 | 5:50 |
| **Hardware IPL** | 4:52 | 4:32 | 2:00 | 2:50 |
| **SLIC MSD IPL with Copy** | 40:34 **#1** | 41:24 **#1** | 8:14 | 6:30 |
| **Shutdown Hardware re-ipl** | 6:16 | 5:31 | 3:22 | 4:10 |
| **SLIC re-ipl** | 8:23 | 12:08 | 2:50 | 2:55 |
| **OS/400** | 20:55 | 20:15 | 2:31 | 2:30 |
| **Total** | 1:49:03 | 1:48:35 | 24:06 | 24:45 |
| **#1:** See section 16.4 for information about systems that have longer C6004250/C6004260 SRC during SLIC MSD IPL | | | | |

MSD is Mainstore Dump. General IPL phase as it relates to the SRCs posted on the operation panel: Processor MSD includes the C1xx xxxx and D1xx xxxx right after the system is forced to terminate. Hardware IPL is the next phase which includes the following group of C1xx xxxx and C3xx xxxx SRCs. 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 C1xx xxxx and C3xx xxxx. SLIC re-IPL follows which are the C600 xxxx SRCs. OS/400 completes with the C900 xxxx SRCs.

## 16.4 MSD Affects on IPL Performance Measurements

SLIC MSD IPL with Copy is greatly affected by the number of dasd units and the data on the system and the jobs executing at the time of the mainstore dump. Most customer systems will see the SRC C6004250 run from 2 min to 30 minutes depending upon the system.

**C6004250/C6004260** - Storage Management directory fix up at C6004250. Under certain conditions the customer could see a switch to C6004260 Storage Management Full Directory Recovery. If a customer systems has a large number of DASD units filled with database files, it can cause this directory fix up to take one to two additional hours over the numbers in our charts. The total time for SLIC MSD IPL with copy portion was 2 hours 40 minutes with a 4 TB database on our system and 40 min without it. The system is in limited paging at this time and can only start so many jobs. If the system has more dasd units than the number of jobs it can start, the other dasd units have to wait until the first jobs have completed before directory recovery can start on them. Whether or not the system will hit this boundary is related to the data on the dasd units and the jobs running in the system at the time of the mainstore dump. A fuzzy line would probably be drawn around 750 arms but a customer could see it a lot lower, if their dasd units have data challenging their capacity.

There is no exact formula to determine if your system will hit this boundary. If it does you have the option to not wait until the system completes. IPL through storage management and copying the dump off to tape if you have a high speed tape drive. The dump is needed by IBM support to help determine what caused the problem on your system in the first place, but you do have the option of exiting out of mainstore without copying the dump. You will still go through a long directory recovery step on this IPL but it will be in full paging and can complete more efficiently.

**DASD Units Effect on MSD Time** - Through some experimental testing we have found that the time spent in MSD copying the data to disk is related to the number of dasd arms available. The following are times with different dasd arms available. These timings are from V4R4 and are for the C6xx 4404 SRC portion of the MSD, not the entire time spent doing the MSD portion of the IPL. C6xx 4404 is the time during the MSD where mainstore is copied to the dasd. By understanding your system configuration, this information and the other information in this document, can help you estimate the amount of time your system may take to IPL when a mainstore dump is needed or happens.

The system used for this test was a 740 270-1513 with 40 GB mainstore and 8 GB dasd arms all RAID protected. The following table shows the effects from varing the number of arms in the ASP where that MSD was copied, and the time it took to complete the MSD.

| Table 16.4.1 | **10 Arms** | **20 Arms** | **36 Arms** | **64 Arms** | **80 Arms** | **112 Arms** | **200 Arms** |
|---|---|---|---|---|---|---|---|
| **40 GB MSD Copy (C600-4404)** | 2 hr 09 min | 1 hr 50 min | 1 hr 07 min | 34 hr | 30 min | 22 min | 13 min |

## 16.5 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 will be a long one, 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 iSeries 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 *iSeries Basic System Operation, Administration, and Problem Handling (SC41-5206)* - or to the redbook *The System Administrator's Companion to iSeries Availability and Recovery (SG24-2161).*

# Chapter 17. Integrated xSeries Server for iSeries

*This chapter gives an introduction to the Integrated xSeries Server, and presents some characteristics and performance impacts for the Integrated xSeries Server on the iSeries.*

## 17.1 Introduction

The Integrated xSeries Server for iSeries (IXS) extends the utility of the iSeries by combining a PC server running Windows NT 4.0 or Windows 2000 with the iSeries. There are several versions of the Integrated xSeries Server:

- The new Integrated xSeries Adapter (IXA) enables SMP IBM xSeries servers to direct attach to the iSeries. The IXA attaches via the iSeries High Speed Link (HSL) bus. The IXA is supported for iSeries models 270 and 8xx and on xSeries 350, 250, Netfinity 7100 and 7600 servers. The xSeries server provides the processors, memory, Server Proven adapters, but no disks. The iSeries provides the disks, storage consolidation and server management. With the IXA, the xSeries server supports larger workloads, more users and greater flexibility to attach devices than other IXS models.
- A new 850 MHz PCI IXS (#2890-002) replaces the 700 MHz PCI IXS (#2890-001) (which is being withdrawn). It is supported on iSeries servers 270, 820, 830, 840, SB2, and SB3.
- The 333 MHz PCI based Integrated Netfinity Server fits in AS/400e series models 170, 150,600, 620, S10, S20, and 720.
- The 333 MHz SPD 'book package' version of the Integrated Netfinity Server fits AS/400e Advanced Series RISC models or integrated expansion units containing book packages.

## 17.2 Configurations

The following illustrations and charts show supported configurations of the IXS and IXAs. In all cases, a separate monitor, keyboard and mouse must be attached to each IXS to act as a Windows console. You may consider using a keyboard-video-mouse switch (KVM) to support multiple IXSs or IXAs. There are many suppliers of this technology which may be found on the Internet. Windows device drivers are provided to share the iSeries's disk, tape, and CD-ROM drives. Integrated xSeries Server operations and systems administration are integrated with the iSeries.

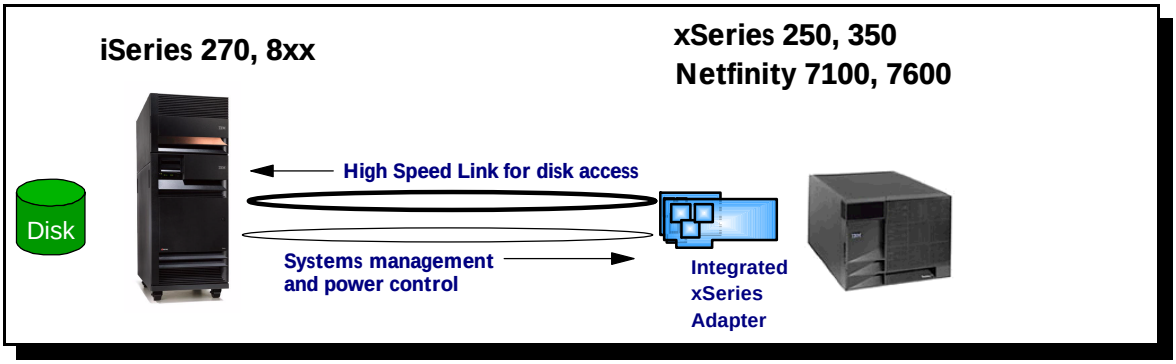



Figure 17.1. Integrated xSeries Adapter, xSeries and iSeries Servers.

- The IXA requires OS/400 Release V5R1.
- Only the Windows 2000 Server family is supported with the IXA.

| iSeries Model | # of Loops | Maximum IXA per loop | Maximum IXA Servers |
|---|---|---|---|
| 270 | 1 | 2 | 2 |
| 820 | 1 | 4 | 4 |
| 830 | 4 | 5 [1] | 8 |
| 840 | 8 | 5 | 16 |
| [1] Only 1 Direct Attach xSeries Server can be place in loop 1 on the model 830. | | | |

Figure 17.2. Integrated xSeries Adapter Maximums

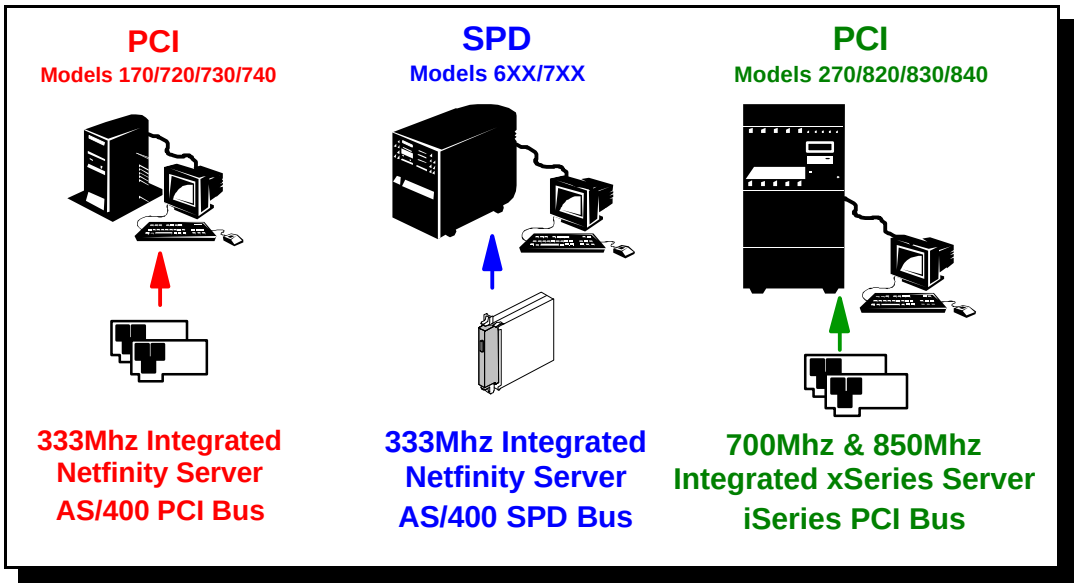



Figure 17.3. 850 MHz PCI, 700 MHz PCI, 333 MHz PCI and SPD Bus Versions of the IXS

For OS/400 V5R1, the maximum number of IXS attached to an iSeries has increased from 16 to 32, depending on the iSeries model.

| iSeries Model | Total Integrated xSeries Servers | Total in main CEC |
|---|---|---|
| 270 | 3 | 1 |
| 820 | 12 | 2 |
| 830 | 28 | 2 |
| 840 | 32 | 2 |
| SB2 | 2 | 2 |
| SB3 | 2 | 2 |
| **Note:** The Total in main CEC number denotes how many Integrated xSeries Servers can be installed in the main system unit. Installing the maximum number of Integrated xSeries Servers may require one or more expansion towers. | | |

Figure 17.4. 850 MHz PCI Maximums

| | Integrated xSeries Server |
|---|---|
| **Processor** | Intel 850 MHz Pentium III processor |
| **Cache** | 256K of on-chip L2 cache |
| **Bus** | 100 MHz front side bus |
| **Memory** | Up to 4GB (max 3712MB addressable)<br>128MB, 256MB, 1GB ECC SDRAMs available |
| **Video** | S3 Savage4 video adapter with 32 MB of video RAM |
| **LAN Adapters** | 1-3 adapters:<br>10/100 Mbps Ethernet<br>1 Gbps Ethernet<br>4/16/100 Mbps Token-Ring |
| **Device Options** | 2 x Universal Serial Bus (USB) ports |
| **Release** | OS/400 V4R5 and above |

Figure 17.5. IXS 850 MHz Server Details

| | PCI Integrated Netfinity Server | Integrated Netfintity Server SPD |
|---|---|---|
| **Processor** | Pentium II 333 Mhz | Pentium II 333 Mhz |
| **Memory** | Up to 1GB | Up to 1GB |
| **iSeries** | AS/400e series with PCI Bus | AS/400e series or Advanced Series with SPD Bus |
| **iSeries Slots** | Pre-reserved | 3 SPD |
| **LAN Adapters** | 1-2 Token-Ring, Ethernet 10/100 | 1-3 Token-Ring, Ethernet 10/100 |
| **Device Options** | Parallel Port 1 Serial Port | Parallel Port 2 Serial Ports |
| **Software Support** | Windows NT Server 4.0 & TSE | Windows NT Server 4.0 & TSE |

Figure 17.6. Integrated Netfinity Server Details

The iSeries Integrated xSeries Server runs Microsoft Windows NT Server Version 4.0, or Windows 2000 Server family; the standard CD-ROM versions that can be purchased from any Microsoft reseller. An IXA attached xSeries Server may only run Windows 2000 Server editions. The Integrated xSeries Server for iSeries and Integrated Netfinity Server for AS/400 have passed the tests to meet Microsoft standards for compatibility with Windows NT Server 4.0 and Windows 2000 Server. See http://www.microsoft.com/hwtest/hcl/. Then do a search on Category: Miscellaneous and Company name: IBM.Microsoft NT Server and Windows 2000 Server have not been modified to run on the Integrated xSeries Server. We have provided device drivers for the Integrated xSeries Server to access the iSeries' disk, tape and CD-ROM drives.

## 17.3 Effects of Windows loads on the iSeries

The Integrated xSeries Server uses iSeries DASD for its hard drives. This is accomplished by special Windows DASD device drivers written for the Integrated xSeries Server. The Windows DASD device drivers cooperate with the iSeries to perform the DASD operations, so iSeries CPU resource is used as well. Thus, IXS and IXA operation primarily effects the DASD subsystem an the iSeries CPU, and is primarily a function of the disk I/O rate.

The following chart shows the measured resource usage of one IXS performing 8k random disk operations to a 4 Gigabyte storage space. The operations are mixed at 67% reads, 33% writes.

| Disk Ops/sec [1] | 400 | 800 | 1,600 | 3,200 |
|---|---|---|---|---|
| HSL (MB/sec) | 3 | 6 | 13 | 26 |
| CPW | 41 | 81 | 152 | 300 |
| Disk arms [2] | 7 | 10 | 18 | 40 |

[1] Disk ops/sec assume 8K byte block transfers, 67% read, 33% write.

[2] Disk arms at 40% utilization (unprotected)

**Performance is 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**.

The above resource guidelines will be the same whatever model of installed IXS or IXA. But, of course, the actual load (in disk ops/sec) will be different for each model and depend heavily on the server application. Also, the number of disk arms required to achieve the same performance levels will change depending on the protection method, such as parity or mirroring, and the size of the storage area being accessed.

To put these numbers in perspective, consider that typical application loads are smaller than the 3,200 disk ops shown in the above table. For example, in lab tests:

- A Netbench 6.0 Enterprise load, running on a 700 MHz 4 way Xeon SMP attached via the IXA resulted in approximately **1000 write ops per second** maximum before the Xeon CPUs become the bottleneck. Thus, for a heavy file serving load, the maximum load produced is about 1k ops/sec.
- An Exchange 5.5 Loadsim test exercising 9000 medium exchange user's produced approximately **400 disk ops per second** - 50/50 reads to writes.
- The IXS and IXA has a similar maximum capacity as a high end raid controller. However, the IXS and IXA capacity is independent of the disk protection. The maximum capacity is similar to that produced by the Netfinity ServeRAID-3H controller, which has a maximum Raid 0 throughput of about 3400 Ops/Sec (Random 8K Byte, 67% read, 33% write operations; according to the Red Book: "Tuning Netfinity Servers for Performance" SG24-5287-01).

## 17.4 Summary

The iSeries Integrated xSeries Server with Windows NT Server or Windows 2000 is a full NT file, print and application server. It provides flexibility for iSeries applications and Windows services in a combination server with improved hardware control, availability, and reduced maintenance costs. The Integrated xSeries Server performs well as a file or application server for popular Windows applications, using the iSeries DASD for its hard drive. As part of the preparation for a combination server installation, care should be taken to estimate the expected workload of the Windows server and reserve iSeries resources for the Integrated xSeries Server.

## 17.5 Additional Sources of Information

***Integrated xSeries Server URL:*** http://www.as400.ibm.com/windowsintegration/
***Microsoft Hardware Compatibility Test URL:*** http://www.microsoft.com/hwtest/hcl
(Catagory:MISC, Company:IBM)

***Redbook:*** "iSeries - Running Windows NT on the Integrated xSeries Server" - SG24-2164 at:
http://www.redbooks.ibm.com/abstracts/sg242164.html

***Redbook:*** "Consolidating Windows 2000 Servers in iSeries: An Implementation Guide for the IBM Integrated xSeries Server for iSeries", SG24-6056 at:
http://www.redbooks.ibm.com/abstracts/SG246056.html

***Redbook:*** "Tuning Netfinity Servers for Performance SG24-5287"
http://www.redbooks.ibm.com/abstracts/SG245287.html

***Online documentation:*** "Windows Server on iSeries"
Go to: http://www.ibm.com/eserver/iseries/infocenter , then, "Network Operating Systems" "Windows Server on iSeries"

# Chapter 18. Logical Partitioning (LPAR)

## 18.1 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.

### 18.1.1 V5R1 additions

With the advent of the V5R1, LPAR provides additional support that includes: dynamic movement of resources without a system or partition reset, processor sharing, and creating a partition using Operations Navigator. For more information on these enhancements, click on System Management at URL: **http://submit.boulder.ibm.com/pubs/html/as400/bld/v5r1/ic2924/index.htm**

With processor sharing, processors no longer have to be dedicated to logical partitions. Instead, a shared processor pool can be defined which will facilitate sharing whole or partial processors among partitions. There is an additional system overhead of approximately 5% (CPU processing) to use processor sharing.

## 18.2 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 D, "AS/400 CPW Values"*.

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.

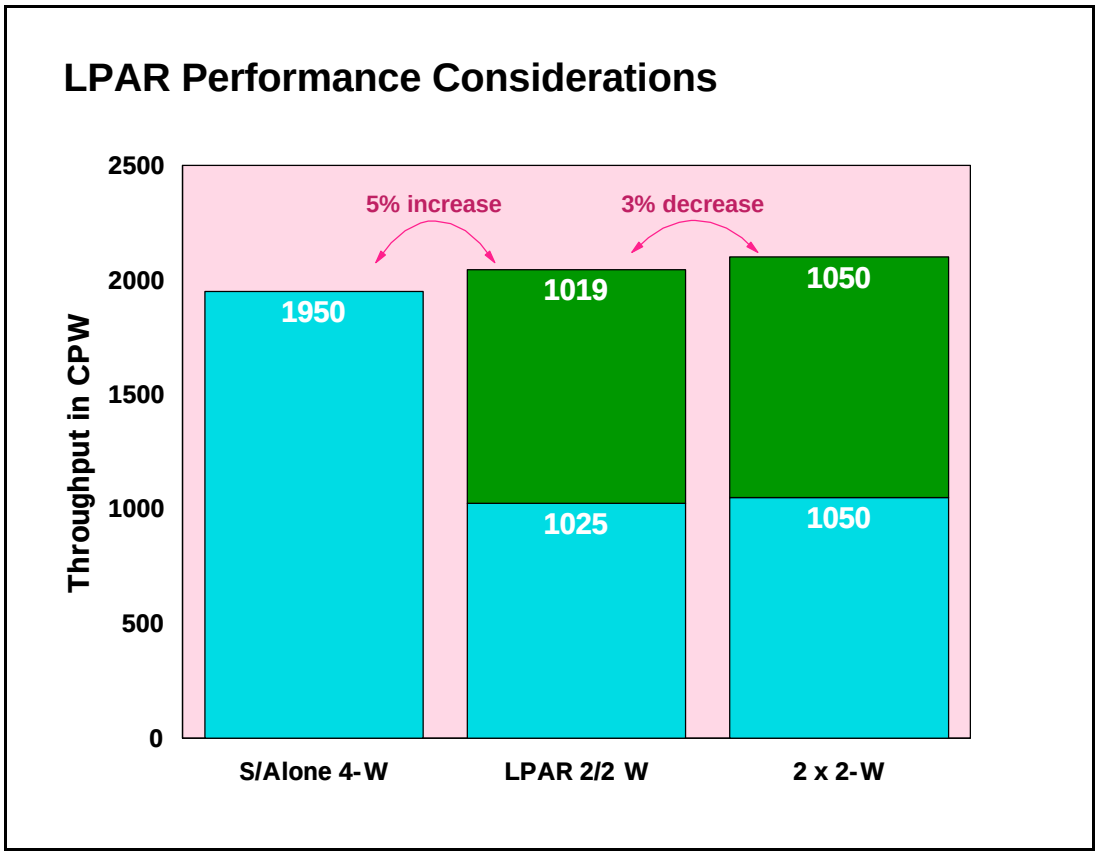



*Figure 18.1. LPAR Performance Measured Against Standalone Systems*

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.

## 18.3 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.

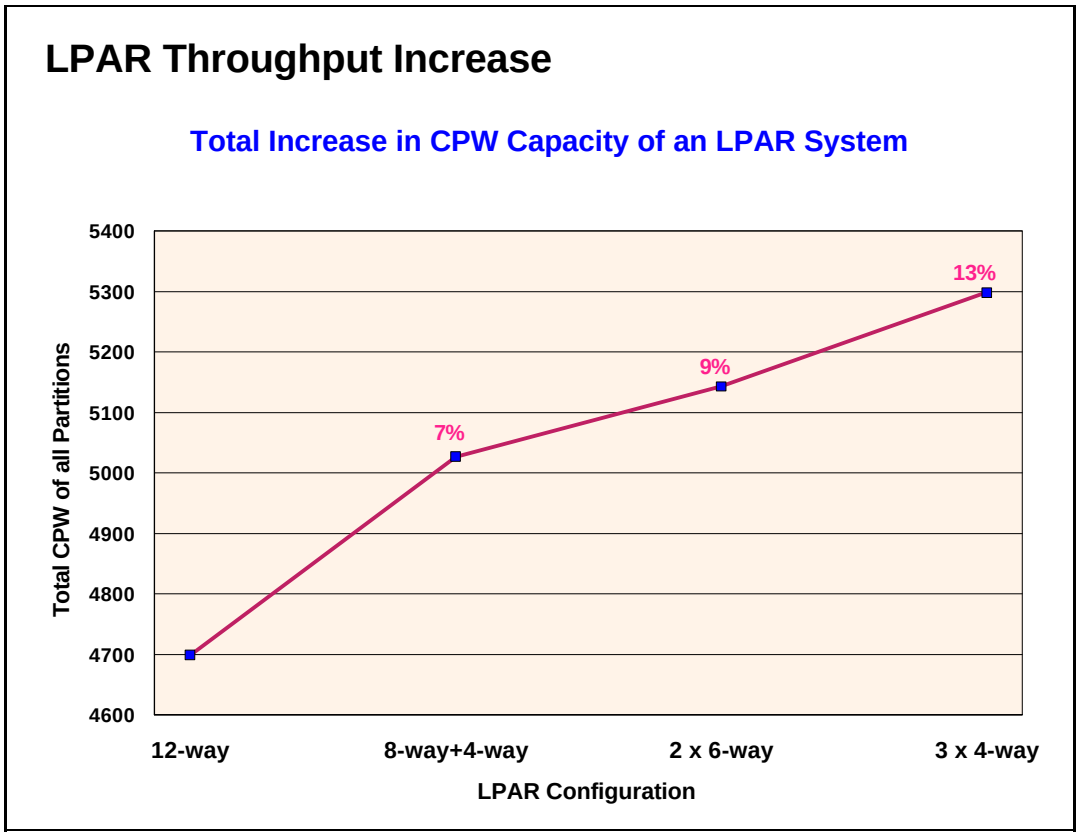



*Figure 18.2. 12 way LPAR Throughput Example*

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.

## 18.4 LPAR Measurements

The following chart shows measurements taken on a partitioned 12-way system with the system's CPU utilized at 70 percent capacity. The system was at the V4R4M0 release level.

Note that the standalone 12-way CPW value of 4700 in our measurement is higher than the published V4R3M0 CPW value of 4550. This is because there was a contention point that existed in the CPW workload when the workload was run on large systems. This contention point was relieved in V4R4M0 and this allowed the CPW value to be improved and be more representative of a customer workload when the workload is run on large systems.

*Table 18.1 12-way system measurements*

| LPAR Configuration | Stand alone 12-way CPW | Total LPAR CPW | CPW Increase | LPAR CPW | | | Average LPAR Overhead |
|---|---|---|---|---|---|---|---|
| | | | | Primary | Secondary | Secondary | |
| 8-way, 4-way | 4700 | 5020 | 7% | 3330 | 1690 | n/a | 10 % |
| (2) 6-ways | 4700 | 5140 | 9% | 2605 | 2535 | n/a | 9 % |
| (3) 4-ways | 4700 | 5290 | 13% | 1770 | 1770 | 1750 | 9 % |

While we saw performance improvements on a 12-way system as shown above, part of those improvements 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 following chart shows our 4-way measurements.

*Table 18.2 4-way system measurements*

| LPAR Configuration | Stand alone 4-way CPW | Total LPAR CPW | CPW Increase | LPAR CPW | | Average LPAR Overhead |
|---|---|---|---|---|---|---|
| | | | | Primary | Secondary | |
| (2) 2-ways | 1950 | 2044 | 5% | 1025 | 1019 | 3 % |

The following chart shows the overhead on n-ways of running a single LPAR partition alone vs. running with other partitions. The differing values for managing partitions is due to the size of the memory nest and the number of processors to manage (n-way size).

*Table 18.3 LPAR overhead per partition*

| Processors | Measured | Projected |
|---|---|---|
| 2 | - | 1.5 % |
| 4 | 3.0 % | - |
| 8 | - | 6.0 % |
| 12 | 9.0 % | - |

The following chart shows projected LPAR capacities for several LPAR configurations. The projections are based on measurements on 1 and 2 way measurements when the system's CPU was utilized at 70 percent capacity. The LPAR overhead was also factored into the projections. The system was at the V4R4M0 release level.

*Table 18.4 Projected LPAR Capacities*

| LPAR Configuration | | Projected LPAR CPW | Projected CPW Increase Over a Standalone 12-way |
|---|---|---|---|
| Number | Processors | | |
| 12 | 1-ways | 5920 | 26 % |
| 6 | 2-ways | 5700 | 21 % |

## 18.5 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 19. Miscellaneous Performance Information

## 19.1 Public Benchmarks (TPC-C, SAP, NotesBench, SPECjbb2000, VolanoMark)

iSeries systems have been represented in several public performance benchmarks. The purpose of these benchmarks is to give an indication of relative strength in a general field of computing. Benchmark results can give confidence in a system's capabilities, but should not be viewed as a sole criterion for the purchase or upgrading of a system. We do not include specific benchmark results in this chapter, because the positioning of these results are constantly changing as other vendors submit their own results. Instead , this section will reference several locations on the internet where current information may be found.

A good source of information on many benchmark results can be found at the ideasInternational benchmark page, at http://www.ideasinternational.com/benchmark/bench.html.

**TPC-C Commercial Performance**

The Transaction Processing Performance Council's TPC Benchmark C (TPC-C (**)) is a public benchmark that stresses systems in a full integrity transaction processing environment. It was designed to stress systems in a way that is closely related to general business computing, but the functional emphasis may still vary significantly from an actual customer environment. It is fair to note that the business model for TPC-C was created in 1990, so computing technologies that were developed in subsequent years are not included in the benchmark.

There are two methods used to measure the TPC-C benchmark. One uses multiple small systems connected to a single database server. This implementation is called a "non-cluster" implementation by the TPC. The other implementation method grows this configuration by coupling multiple database servers together in a clustered environment. The benchmark is designed in such a way that these clusters scale far better than might be expected in a real environment. Less than 10% of the transactions touch more than one of the database server systems, and for that small number the cross-system access is typically for only a single record. Because the benchmark allows unrealistic scaling of clustered configurations, we would advise against making comparisons between clustered and non-clustered configurations. All iSeries results and AS/400 results in this benchmark are non-clustered configurations - showing the strengths of our system as a database server.

The most current level of TPC-C benchmark standards is Version 5, which requires the same performance reporting metrics but now requires pricing of configurations to include 24 hr x 7 day a week maintenance rather than 8 hr x 5 day a week and some additional changes in pricing the communication connections. All previous version submissions from reporting vendors have been offered the opportunity to simply republish their results with these new metric ground rules. And as of April, 2001 not all vedors have chosen to republish their results to the new Version 5 stardard. iSeries and pSeries has republished.

For additional information on the benchmark and current results, please refer to the TPC's web site at: http://www.tpc.org

## SAP Performance Information

Several Business Partner companies have defined benchmarks for which their applications can be rated on different hardware and middle ware platforms. Among the first to do this was SAP. SAP has defined a suite of "Standard Application Benchmarks", each of which stresses a different part of SAP's solutions. The most commonly run of these is the SAP-SD (Sales and Distribution) benchmark. It can be run in a 2-tier environment, where the application and database reside on the same system, or on a 3-tier environment, where there are many application servers feeding into a database server.

Care must be taken to ensure that the same level of software is being run when comparing results of SAP benchmarks. Like most software suppliers, SAP strives to enhance their product with useful functions in each release. This can yield significantly different performance characteristics between releases such as 4.0B, 4.5B, and 4.6C. It should be noted that, although SAP is used as an example here, this situation is not restricted to SAP software.

For more information on SAP benchmarks, go to http://www.sap.com and process a search for Standard Application Benchmarks Published Results.

## NotesBench

There are several benchmarks that are called "Notesbench xxx". All come from the Notesbench Consortium, a consortium of vendors interested in using benchmarks to help quantify system capabilities using Lotus Domino functions. The most popular benchmark is Notesbench R5 Mail, which is actually a mail and calendar benchmark that was designed around the functions of Lotus Domino Release 5.0. AS/400 and iSeries systems have traditionally demonstrated very strong performance in both capacity and response time in Notesbench results.

For official iSeries audited NotesBench results, see http://www.notesbench.org . (Note: in order to access the NotesBench results you will need to apply for a userid/password through the Notesbench organization. Click on Site Registration at the above address.) An alternate is to refer to the ideasInternational web site listed above.

For more information on iSeries performance in Lotus Domino environments, refer to Chapter 11 of this document.

## SPECjbb2000

The Standard Performance Evaluation Corporation (SPEC) defined, in June, 2000, a server-side Java benchmark called SPECjbb2000. It is one of the only Java-related benchmarks in the industry that concentrates on activity in the server, rather than a single client. The iSeries architecture is well suited for an object-oriented environment and it provides one of the most efficient and scalable environments for server-side Java workloads. iSeries and AS/400 results are consistently at or near the top rankings for this benchmark.

For more information on SPECjbb2000 and for published results, see http://www.spec.org/osg/jbb2000/

For more information on iSeries performance in Java environments, refer to Chapter 7 of this document.

## VolanoMark

IBM has chosen the VolanoMark benchmark as another means for demonstrating strength with server-side Java applications. VolanoMark is a 100% Pure Java server benchmark characterized by long-lasting network connections and high thread counts. It is as much a test of tcp/ip strengths as it is of multithreaded, server-side Java strengths. In order to scale well in this benchmark, a solution needs to scale well in tcp/ip, Java-based applications, multithreaded application, and the operating system in general. Additional information on the benchmark can be found at http://www.volano.com/benchmarks.html. This web site is primarilly focused on results for systems that the Volano company measures themselves. These results tend to be for much smaller, Intel-based systems that are not comparable with iSeries servers. The web site also references articles written by other groups regarding their measurements of the benchmark, including AS/400 and iSeries articles. iSeries servers have demonstrated significant strengths in this benchmark, particularly in scaling to large systems.

## 19.2 Dynamic Priority Scheduling

On an AS/400 CISC-model, all ready-to-run OS/400 jobs and Licensed Internal Code (LIC) tasks are sequenced on the Task Dispatching Queue (TDQ) based on priority assigned at creation time. In addition, for N-way models, there is a cache affinity field used by Horizontal Licensed Internal Code (HLIC) to keep track of the processor on which the job was most recently active. A job is assigned to the processor for which it has cache affinity, unless that would result in a processor remaining idle or an excessive number of higher-priority jobs being skipped. The priority of jobs varies very little such that the resequencing for execution only affects jobs of the same initially assigned priority. This is referred to as Fixed Priority Scheduling.

For V3R6 and beyond, the new algorithm being used is Dynamic Priority Scheduling. This new scheduler schedules jobs according to "delay costs" dynamically computed based on their time waiting in the TDQ as well as priority. The job priority may be adjusted if it exceeded its resource usage limit. The cache affinity field is no longer used in a N-way multiprocessor machine. Thus, on an N-way multiprocessor machine, a job will have equal affinity for all processors, based only on delay cost.

A new system value, QDYNPTYSCD, has been implemented to select the type of job dispatching. The job scheduler uses this system value to determine the algorithm for scheduling jobs running on the system. The default for this system value is to use Dynamic Priority Scheduling (set to '1'). This scheduling scheme allows the CPU resource to be spread to all jobs in the system.

The benefits of Dynamic Priority Scheduling are:

- No job or set of jobs will monopolize the CPU
- Low priority jobs, like batch, will have a chance to progress
- Jobs which use too much resource will be penalized by having their priority reduced
- Jobs response time/throughput will still behave much like fixed priority scheduling

By providing this type of scheduling, long running, batch-type interactive transactions, such as a query, will not run at priority 20 all the time. In addition, batch jobs will get some CPU resources rather than interactive jobs running at high CPU utilization and delivering response times that may be faster than required.

To use Fixed Priority Scheduling, the system value has to be set to '0'.

## Delay Cost Terminology

- Delay Cost

  Delay cost refers to how expensive it is to keep a job in the system. The longer a job spends in the system waiting for resources, the larger its delay cost. The higher the delay cost, the higher the priority. Just like the priority value, jobs of higher delay cost will be dispatched ahead of other jobs of relatively lower delay cost.

- Waiting Time

  The waiting time is used to determine the delay cost of a job at a particular time. The waiting time of a job which affects the cost is the time the job has been waiting on the TDQ for execution.

- Delay Cost Curves

  The end-user interface for setting job priorities has not changed. However, internally the priority of a job is mapped to a set of delay cost curves (see "Priority Mapping to Delay Cost Curves" below). The delay cost curve is used to determine a job's delay cost based on how long it has been waiting on the TDQ. This delay cost is then used to dynamically adjust the job's priority, and as a result, possibly the position of the job in the TDQ.

  On a lightly loaded system, the jobs' cost will basically stay at their initial point. The jobs will not climb the curve. As the workload is increased, the jobs will start to climb their curves, but will have little, if any, effect on dispatching. When the workload gets around 80-90% CPU utilization, some of the jobs on lower slope curves (lower priority), begin to overtake jobs on higher slope curves which have only been on the dispatcher for a short time. This is when the Dynamic Priority Scheduler begins to benefit as it prevents starvation of the lower priority jobs. When the CPU utilization is at a point of saturation, the lower priority jobs are climbing quite a way up the curve and interacting with other curves all the time. This is when the Dynamic Priority Scheduler works the best.

  Note that when a job begins to execute, its cost is constant at the value it had when it began executing. This allows other jobs on the same curve to eventually catch-up and get a slice of the CPU. Once the job has executed, it "slides" down the curve it is on, to the start of the curve.

## Priority Mapping to Delay Cost Curves

The mapping scheme divides the 99 'user' job priorities into 2 categories:

- User priorities 0-9

  This range of priorities is meant for critical jobs like system jobs. Jobs in this range will NOT be overtaken by user jobs of lower priorities. NOTE: You should generally not assign long-running, resource intensive jobs within this range of priorities.

- User priorities 10-99

This range of priorities is meant for jobs that will execute in the system with dynamic priorities. In other words, the dispatching priorities of jobs in this range will change depending on waiting time in the TDQ if the QDYNPTYSCD system value is set to '1'.

The priorities in this range are divided into groups:

- Priority 10-16
- Priority 17-22
- Priority 23-35
- Priority 36-46
- Priority 47-51
- Priority 52-89
- Priority 90-99

Jobs in the same group will have the same resource (CPU seconds and Disk I/O requests) usage limits. Internally, each group will be associated with one set of delay cost curves. This would give some preferential treatment to jobs of higher user priorities at low system utilization.

With this mapping scheme, and using the default priorities of 20 for interactive jobs and 50 for batch jobs, users will generally see that the relative performance for interactive jobs will be better than that of batch jobs, without CPU starvation.

## Performance Testing Results

Following are the detailed results of two specific measurements to show the effects of the Dynamic Priority Scheduler:

In Table 19.1, the environment consists of the RAMP-C interactive workload running at approximately 70% CPU utilization with 120 workstations and a CPU intensive interactive job running at priority 20.

In Table 19.2 below, the environment consists of the RAMP-C interactive workload running at approximately 70% CPU utilization with 120 workstations and a CPU intensive batch job running at priority 50.

*Table 19.1. Effect of Dynamic Priority Scheduling: Interactive Only*

| | **QDYNPTYSCD = '1' (ON)** | **QDYNPTYSCD = '0'** |
|---|---|---|
| Total CPU Utilization | 93.9% | 97.8% |
| Interactive CPU Utilization | 77.6% | 82.2% |
| RAMP-C Transations per Hour | 60845 | 56951 |
| RAMP-C Average Response Time | 0.32 | 0.75 |
| Priority 20 CPU Intensive Job CPU | 21.9% | 28.9% |

*Table 19.2. Effect of Dynamic Priority Scheduling: Interactive and Batch*

| | **QDYNPTYSCD = '1' (ON)** | **QDYNPTYSCD = '0'** |
|---|---|---|
| Total CPU Utilization | 89.7% | 90.0% |
| Interactive CPU Utilization | 56.3% | 57.2% |
| RAMP-C Transations per Hour | 61083 | 61692 |
| RAMP-C Average Response Time | 0.30 | 0.21 |
| Batch Priority 50 Job CPU | 15.0% | 14.5% |
| Batch Priority 50 Job Run Time | 01:06:52 | 01:07:40 |

**Conclusions/Recommendations**

- When you have many jobs running on the system and want to ensure that no one CPU intensive job 'takes over' (see Table 19.1 above), Dynamic Priority Scheduling will give you the desired result. In this case, the RAMP-C jobs have higher transaction rates and faster response times, and the priority 20 CPU intensive job consumes less CPU.

- Dynamic Priority Scheduling will ensure your batch jobs get some of the CPU resources without significantly impacting your interactive jobs (see Table 96). In this case, the RAMP-C workload gets less CPU utilization resulting in slightly lower transaction rates and slightly longer response times. However, the batch job gets more CPU utilization and consequently shorter run time.

- It is recommended that you run with Dynamic Priority Scheduling for optimum distribution of resources and overall system performance.

For additional information, refer to the *Work Management Guide*.

---

## 19.3 Main Storage Sizing Guidelines

To take full advantage of the performance of the new AS/400 Advanced Series using PowerPC technology, larger amounts of main storage are required. To account for this, the new models are provided with substantially more main storage included in their base configurations. In addition, since more memory is required when moving to RISC, memory prices have been reduced.

The increase in main storage requirements is basically due to two reasons:

- When moving to the PowerPC RISC architecture, the number of instructions to execute the same program as on CISC has increased. This does not mean the function takes longer to execute, but it does result in the function requiring more main storage. This obviously has more of an impact on smaller systems where fewer users are sharing the program.

- The main storage page size has increased from 512 bytes to 4096 bytes (4KB). The 4KB page size is needed to improve the efficiency of main storage management algorithms as main storage sizes increase dramatically. For example, 4GB of main storage will be available on AS/400 Advanced System model 530.

  The impact of the 4KB page size on main storage utilization varies by workload. The impact of the 4KB page size is dependent on the way data is processed. If data is being processed sequentially, the 4KB page size will have little impact on main storage utilization. However, if you are processing data randomly, the 4KB page size will most likely increase the main storage utilization.

## 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

Due to the large range of AS/400 processors and due to an ever increasing variance in the complexity of user applications, paging guidelines for user pools are no longer published. Only machine pool guidelines and system wide guidelines (sum of faults in all the pools) are published. Even the system wide guidelines are just that...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 AS/400 NetFinity Capacity Planning

Performance information for AS/400 NetFinity attached to a V4R1 AS/400 is included below. The following NetFinity functions are included:

- Time to collect software inventory from client PCs
- Time to collect hardware inventoy from client PCs

The figures below illustrate the time it takes to collect software and hardware inventory from various numbers of client PCs. This test was conducted using the Rochester development site, during normal working hours with normal activity (ie. not a dedicated environment). This environment consists of:

- 16 and 4Mb token ring LANs (mostly 16)
- LANs connected via routers and gateways
- Dedicated AS/400
- TCP/IP
- Client PCs varied from 386s to Pentiums (mostly 100 MHz with 32MB memory), using OS/2, Windows/95 and NT
- About 20K of data was collected, hardware and software, for each client

While these tests were conducted in a typical work envirnoment, results from other environments may vary sigificantly from what is provided here.

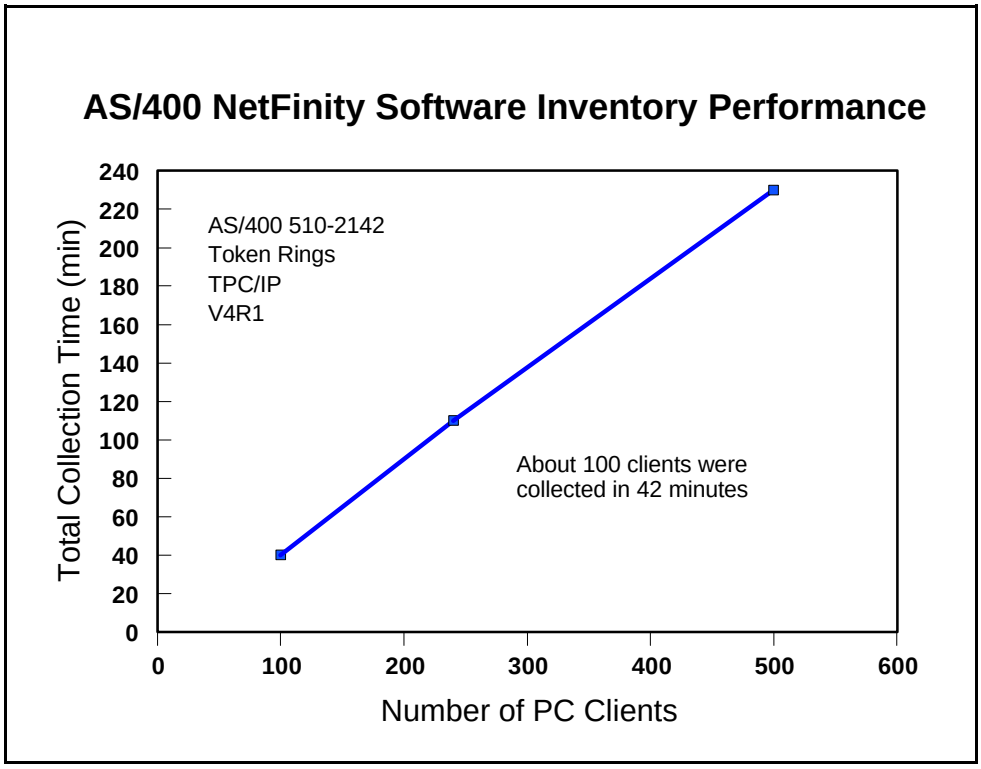



Figure 19.1. AS/400 NetFinity Software Inventory Performance

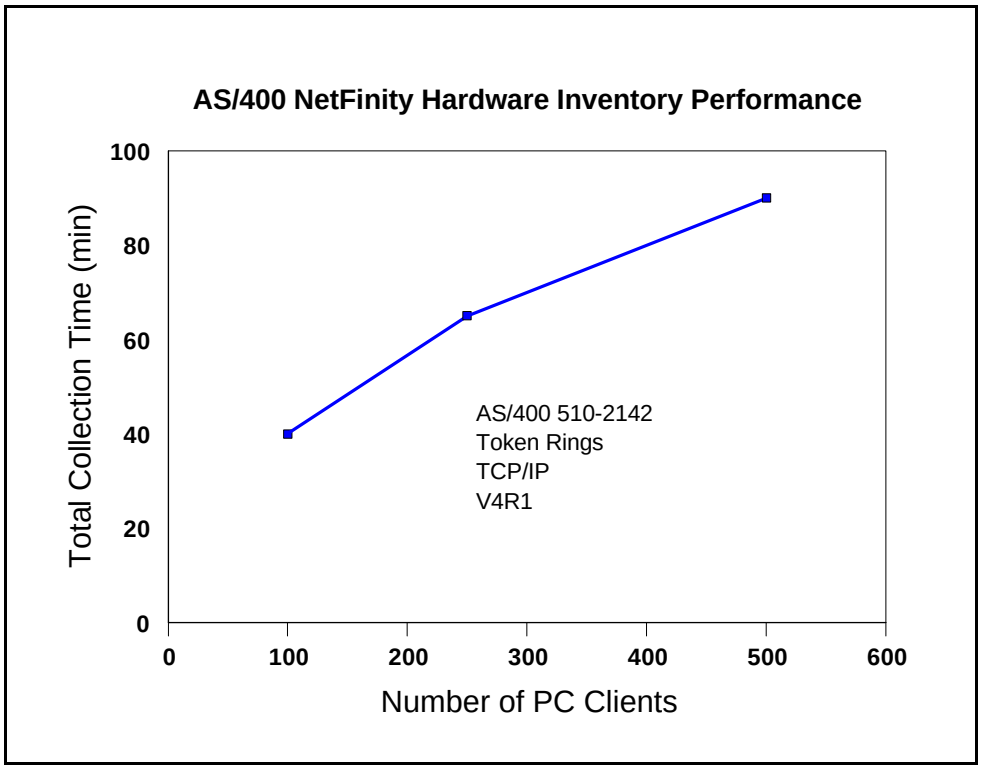



Figure 19.2. AS/400 NetFinity Hardware Inventory Performance

**Conclusions/Recommendations for NetFinity**

1. The time to collect hardware or software information for a number of clients is fairly linear.

2. The size of the AS/400 CPU is not a limitation. Data collection is performed at a batch priority. CPU utilization can spike quite high (ex. 80%) when data is arriving, but in general is quite low (ex. 10%).

3. The LAN type (4 or 16Mb Token Ring or Ethernet) is not a limitation. Hardware collection tends to be more chatty on the LAN than software collection, depending on the hardware features.

4. The communications protocol (IPX, TCP/IP, or SNA) is not a limitation.

5. Collected data is automatically stored in a standard DB/2/400 database file, accessible by SQL and other APIs.

6. Collection time depends on clients being powered-on and the needed software turned on. The server will retry 5 times.

7. The number of jobs on the server increases during collection and decreases when not needed.

# Chapter 20. General Performance Tips and Techniques

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 the AS/400. It may also contain some general guidelines.

## 20.1 Adjusting Your Performance Tuning for Threads

### History

Historically, the AS/400 programmer has 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 AS/400 subsystems (e.g. QBATCH, QINTER, MYOWNSUBSYSTEM, etc.) have not had to think much about the distinction between a "job" and a "thread."

### The Coming Change

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 AS/400 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.

## 20.2 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 surpisingly often irrelevant in actual cases, many languages have specific definitions that allow lattitude 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.

### LICOPT

A new option has been added to most ILE Languages called LICOPT. This allows language specific optimizations to be turned on and off as individual items. A full description of this is well beyond the

scope of this paper, but those interested in the highest level of performance and yet minimizing potential difficulties with specific optimization types would do well to study these options.

---

## 20.3 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 trade-offs 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 free-standing 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 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 ith 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?

# Chapter 21. AS/400 PASE Performance

## 21.1 Introduction

AS/400 Portable Application Solutions Environment (AS/400 PASE) is designed to expand the AS/400 platform solutions portfolio by allowing customers and software vendors to port existing AIX applications to the AS/400 with minimal effort.

AS/400 PASE is an integrated runtime environment for AIX applications running on the AS/400 system. As a native runtime it does not suffer the drawbacks of an emulation environment.

AS/400 PASE is designed to accept direct ports from AIX. AS/400 PASE relies on the AIX Application Binary Interface, so ports from other UNIX(tm) environments will require an initial port to AIX to ensure compatibility.

Until recently, the AS/400 Integrated Language Environment accounted for the majority of C and C++ application ports, many of which originally ran on UNIX. Recently, however, the AS/400 system has focused on Java and Domino based applications, opening up new application porting and modernization opportunities for solutions developers and allowing solution developers another option for rapidly porting UNIX applications to the AS/400.

AS/400 PASE enables the AS/400 to expand its solutions portfolio, focusing on specific industry and application segments. For example, new supply chain management solutions that integrate with ERP applications are targeted at industrial and distribution industries. Certain applications may fit better in the ILE while others will fit better in AS/400 PASE.

AS/400 PASE is supported on all AS/400e series servers (AS/400 systems introduced after 8/97). AS/400 PASE takes advantage of the AS/400 system and RS/6000's common investment in PowerPC processor technology. The PowerPC processor switches from its normal AS/400 mode, in order to execute an application in the AS/400 PASE runtime. Applications running in AS/400 PASE may need to be enabled to access DB2 Universal Database for AS/400 and integrated with AS/400 security and operations, such as backup. The ease of porting depends on the APIs used by the application. Some binaries will run without change, while others may require minor to substantial modifications.

Application providers can obtain more support for porting their applications to AS/400 PASE from PartnerWorld for Developers at http://www.as400.ibm.com/developer/ .

### Linux and PASE

In V5R1, iSeries has added support for Linux under LPAR. Just by adding another choice, developers might wonder where to target their next ported application: Linux, PASE, or ILE?

If the application runs on AIX today, PASE is probably still the best choice for porting. PASE runs on more models and feature codes of iSeries and the predecessor AS/400. PASE is also generally slightly faster, at least for application code, as xlc seems to out-optimize Linux' gcc code generation in most cases. Linux will likewise be favored if the application is already coded for some form of Linux. For applications

that are more platform neutral, have special performance considerations, or which have multiple source code versions making the choice less obvious, see the Linux chapter of this document for further advise on all three choices (PASE, Linux, ILE).

**AS/400 PASE Technical Overview**

AS/400 PASE is a program-execution environment on the AS/400 system that provides a traditional memory model (not single-level store) and allows direct access to machine instructions (without the mapping of MI architecture). Programs running in AS/400 PASE have direct access to the full capabilities of the user-state architecture of PowerPC, augmented by system services to interoperate with the Single-Level Store (SLS) environment.

In the single-level store environment, all processes share a single address space that provide a mapping for all memory in the system (except for unnamed Teraspace regions). Security and integrity are provided through a combination of a). page-level hardware storage protection and b). controlling hardware instruction sequences (so that only “safe” memory addresses are generated). Hardware instruction sequences are controlled by requiring all programs to be generated by the System Licensed Internal Code (SLIC) translator from a high-level program description called an MI program template.

AS/400 PASE provides a separate private address space for each process and limits the mappings in each private address space to only memory that is “safe” for access by user programs. Programs running in AS/400 PASE can only address memory that is mapped into the address space where the program runs and do not have direct access to the special PowerAS instructions that build tagged MI pointers. In this way, a AS/400 PASE program can run any arbitrary sequence of (user-state PowerPC) hardware instructions without jeopardizing system security or integrity.

The AS/400 PASE address space provides a mapping from AS/400 PASE addresses to addresses in a Teraspace region. Any memory mapped into the AS/400 PASE region of Teraspace has exactly the same accessibility (relative address and storage protection) to both AS/400 PASE programs and SLS programs. Both SLS segments and unnamed memory can be mapped into Teraspace. AS/400 PASE and SLS can share memory (but in a controlled manner, limited to the memory mapped into the private address space), and programs can call back and forth between environments (within the context of a single process). AS/400 PASE programs deal with 4-byte (untagged) pointers, in contrast to the 16-byte tagged MI pointers used in SLS.

AS/400 PASE currently provides support for 32-bit AIX application, in support of a 32-bit addressing model. The processors used in AS/400 systems also support 64-bit applications.

**AS/400 PASE Run-time Support**

V4R5 adds significant support including Xwindows and a large number of shells and utilities.

Applications running in AS/400 PASE work in ASCII. The AIX C compiler (xlc) does not support EBCDIC. Any AS/400 PASE runtime service the system provides handles ASCII/EBCDIC conversions as needed, although generally no conversions are done for data read or written to a file descriptor (bytestream file or socket).

AS/400 PASE programs pass ASCII (or UTF-8) path names to the open function to open bytestream files, but any data read or written from the open file is unconverted. It is the responsibility of the application

running in AS/400 PASE to handle character encoding conversions for calls from AS/400 PASE to arbitrary ILE procedures. AS/400 PASE runtime support includes functions iconv_open, iconv, and iconv_close for character encoding conversion.

AS/400 PASE runtime only provides stream file access to IFS, which is the equivalent of SYSIFCOPT(*IFSIO) for ILE C code. Access to DB2/400 database is provided through SQL CLI functions exported by an AS/400 PASE shared library included with OS/400 option 33. A AS/400 PASE program that requires access to object types and/or interface options not supported through IFS or SQL CLI can directly call ILE procedures.

Environment variable support for AS/400 PASE runs independently of SLS runtime. The system does not implicitly set any AS/400 PASE environment variables, but the Qp2RunPase API allows the caller to specify a set of environment variables to initialize in AS/400 PASE, and program QP2SHELL passes a copy of all ILE environment variables to the AS/400 PASE program. SLIC implements support for the (AIX) system calls needed to run the C library runtime subset supported by AS/400 PASE, and also supports system calls for platform-specific functions such as building a tagged space pointer (_SETSPP) and calling an ILE procedure (_ILECALL).

**AS/400 PASE Development Environment**

AS/400 PASE development requires an AIX system to run the compiler (xlc, xlC, or some other AIX compiler) and linker (ld command). The AIX assembler for PowerPC can also be used. Application development for any release of AS/400 PASE must be done on an AIX system release that is compatible (and optimally the same as) the AIX release for which the AS/400 PASE release provides equivalence. For example, V4R5 AS/400 PASE is based on a subset of AIX 4.3.3.

**Characteristics of Application Candidates for AS/400 PASE**

When planning to port an application from AIX to the AS/400 there are three choices: you can port to the AS/400 Integrated Language Environment (ILE) , Linux on iSeries under LPAR, or you can port to AS/400 PASE.

Here are some of the reasons to choose AS/400 PASE:

1. If the UNIX/AIX APIs your applications uses are already supported by AS/400 PASE, then there is very little application porting to be done. You can determine how well your application conforms to supported AS/400 PASE APIs by using the “filtering” tool found at http://www.as400.ibm.com/developer/porting/apitool.html .
2. AS/400 PASE provides an environment for running more computationally intensive applications on the AS/400 system by providing optimized math libraries.
3. AS/400 PASE allows the use of UNIX based build processes, which is especially useful when you have an existing, complicated build process.
4. AS/400 PASE supplies support for *fork* and *exec*, which does not currently exist on the AS/400 system (except through *spawn* which is significantly different).
5. AS/400 PASE is designed to satisfy dependencies on an ASCII character set and to satisfy dependencies on X-Windows support.
6. AS/400 PASE fully supports ANSII C, C++ and FORTRAN.
7. Shell programming is supported.
8. Better overall application performance (over Linux). Note: Code that invokes significant operating system services may make this factor less important.

It should be noted that most of these advantages are shared, in broad brush, with Linux (except best overall application performance). Applications originally coded for AIX probably belong in PASE. If other considerations apply, see the Linux chapter for further advise about choosing PASE, ILE, or Linux.

## 21.2 V4R5 Performance Test Results

A number of performance related tests have been conducted to compare the performance of AS/400 PASE to other environments on AS/400 and to compare performance to similarly configured (especially CPU MHz) RS/6000s running the application in an AIX environment.

The tests were deliberately chosen to represent a diverse set of applications; both simple tests using basic primitives and more complex tests using subsets of real commercial applications. These include subject areas such as CPU intensive workloads, forking, DB2 Command Language Interface (CLI) workload, I/O to the Integrated File System (IFS), cross environment calls, network and socket performance and a ported commercial application.

In some instances, it is relevant to compare AS/400 PASE to AIX and in other instances to compare AS/400 PASE to the AS/400 Integrated Language Environment (ILE). Ease of porting is an important consideration. When porting from a UNIX environment to AS/400, the software developer is now given two options, either to port to ILE or to use AS/400 PASE. The difference between ILE and AS/400 PASE is fairly small, assuming that above guidelines are followed. AS/400 PASE may be the preferred option, for example, in computationally intensive applications.

### CPU Intensive Workloads

While most applications which run on AS/400 are commercial by nature, many modern applications have the characteristic of requiring additional CPU capacity. In order to measure the relative performance of AS/400 PASE, two workloads were devised that perform numeric intensive calculations. Be aware that these results cannot be used to compare AS/400 PASE directly to other platforms using other numeric workloads such as the SPEC series of workloads.

Of the two workloads, one represented integer arithmetic and the second represented floating point operations. They were run on an AS/400 and an RS/6000 which had CPUs designed with the same technology and running at the same MHz. The test code resides in main storage and does relatively small amounts of I/O.

Results are shown in the Table 21.1 (note: a higher rating is better):

| OVERALL | | | | |
|---|---|---|---|---|
| | # | CPU | Integer | Float |
| Machine | CPU | MHz | rating | rating |
| *RS/6000* | 1 | 262 | 796 | 1642 |
| | | | | |
| *AS/400* | 1 | 262 | 767 | 1505 |
| *HMT OFF* | | | | |
| *% Difference* | | | -3.73% | -8.36% |

*Table 21.1 Comparison AS/400 PASE to AIX for CPU intensive workloads*

The RS/6000 does not have hardware multitasking (HMT) enabled, so this feature was turned off on the AS/400 prior to running the test. This was a single user test and in general a single task will run faster with HMT turned off. However, in a multi-user environment, HMT turned on will often improve CPU utilization by 10-20%. As is the case with all performance recommendations, results vary in different customer environments, so test this feature before using it.

As can be seen in the table above, for CPU intensive workloads, AS/400 PASE has similar performance, albeit a little slower, than a similarly configured RS/6000.

**Forking Performance**

A simple workload was used to test the CPU overhead in AS/400 PASE forking compared to AIX forking. The test was run on an AS/400 and an RS/6000, both with CPU processor cycle times of 262 MHz.

The test created a variable number of forked jobs which once created went into a sleep state. In the AS/400 PASE implementation, spawning a new process requires considerable OS/400 initialization, whereas AIX spawns a new process more efficiently. The actual cost per fork on the AS/400 was about 5mS (on the above AS/400 system). The comparable cost per fork on AIX was about 0.3mS. Although the AS/400 overhead was somewhat higher than its AIX counterpart, this cost may be insignificant when compared to the total work being done during each transaction and may be acceptable for most commercial applications.

The added value of AS/400 PASE is to easily enable forking as part of a port, and provided the application minimal forking (which is usually the case), then the application will see minimal changes in performance.

**Networking Testing**

The NetPerf workload is a primitive-level function workload used to explore communications performance. It consists of C programs that run between a client AS/400 and a server AS/400. Multiple instances of NetPerf can be executed over multiple connections to increase the system load. The programs communicate with each other using sockets or SSL programming APIs.

Whereas most 'real' application programs will process data in some fashion, 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 database accesses). Figure 21.1 shows the relative performance when running NetPerf in AS/400 PASE and ILE.

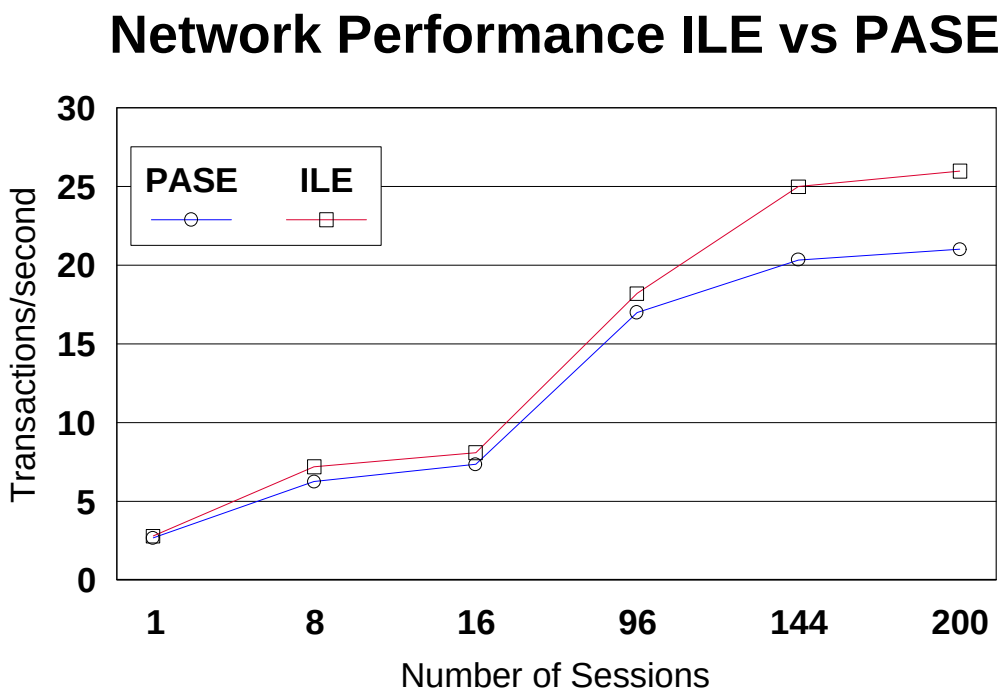



*Figure 21.1 Comparison of Network Performance for AS/400 PASE and ILE*

The performance of AS/400 PASE and ILE track very closely until about 96 sessions. The number of instructions per NetPerf transaction is slightly higher for PASE, with the throughput for both AS/400 PASE and ILE being limited by CPU capacity.

**Cross Environment Calls**

This test was created to determine the overhead of calling ILE functions from AS/400 PASE. The AS/400 PASE environment is 32-bit and ILE is 64-bit. This requires an address translation between the two address spaces.

In order to test the call overhead from AS/400 PASE, a number of scenarios were used which varied the number and type of parameters. Some parameters are passed by value whereas others use pointers.

Results (on an AS/400 4-way model 730-2067 with #1511 feature) are shown in the Figure 21.2.

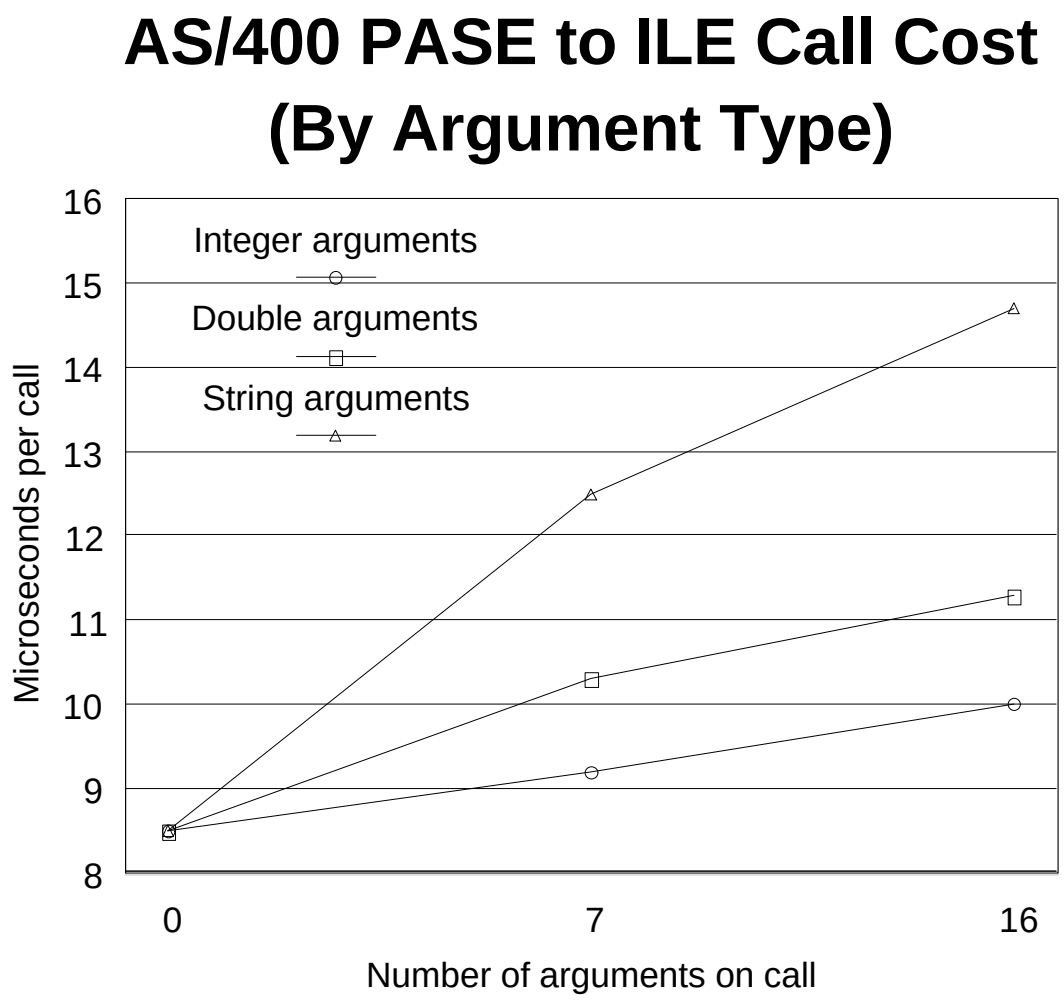



*Figure 21.2 Cross environment calls comparing AS/400 PASE to ILE and ILE to ILE*

As Figure 21.2 shows, the cost of calling ILE from an AS/400 PASE program depends on a number of factors including the number and type of arguments. The cost of calling ILE from AS/400 PASE is longer than ILE to ILE bound calls. The impact of the cost of calling ILE from AS/400 PASE will be a function of how much processing in the AS/400 PASE app is done leading up to the call, how much processing is done in the ILE code, and how frequently the ILE code is called.

Figure 21.3 shows the flow of control from an AS/400 PASE program to ILE code. If the time in the AS/400 PASE program leading up to the call combined with time spent in the ILE code is large compared to the cost of the call itself, then the impact of calling ILE will be largely irrelevant even if ILE is called many times. However, if very little time is spent in the AS/400 PASE program and ILE code then the cost of the call itself will be more of an impact to overall performance, especially if the ILE call is made numerous times.

# AS/400 PASE to ILE Call Diagram

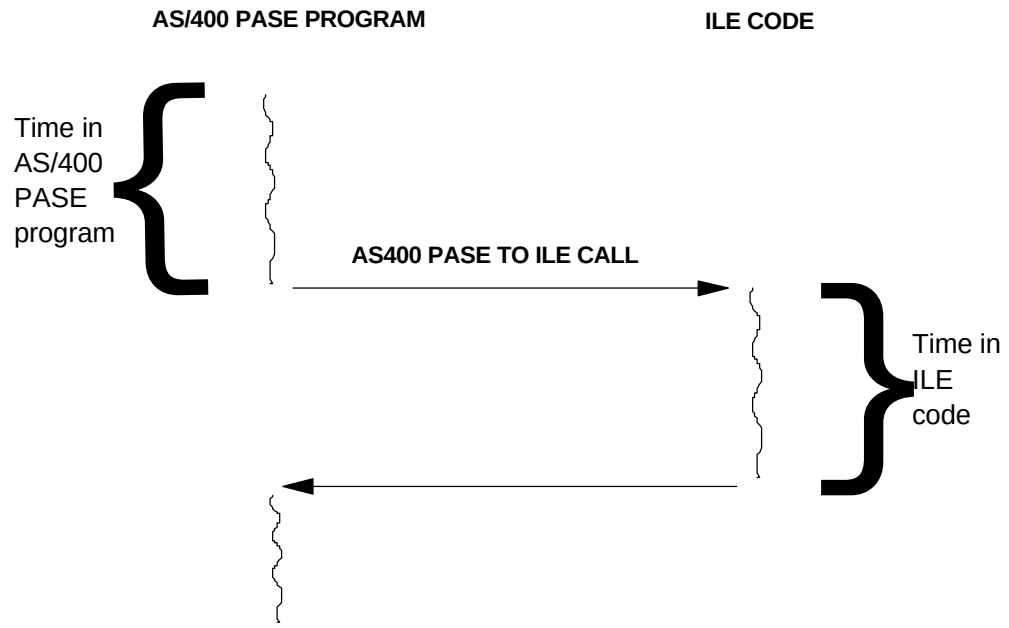



Figure 21.3 AS/400 PASE to ILE Call Flow

**DB2/400 CLI Performance Testing**

AS/400 PASE provides a shared library that enables access to the native AS/400 DB2 Universal Database (UDB) Command Language Interface (CLI).

In order to test the performance of this interface, a workload was built which accessed a database table randomly, alternating between a random read and a random read followed by an update. The file used contained 100,000 records.

This test uses a number of the common CLI APIs to demonstrate in a sample application, that the difference in performance is relatively small in these two environments, and should not be a major issue when choosing which environment to use when porting.

The testing was done on an AS/400 4-way  model 730-2067 with #1511 feature. Results of running this workload are shown in Table 21.2.

| PASE CLI tests on AS/400 730 4W N*CPW=2000 | | | |
|---|---|---|---|
| (Elapsed time in seconds for 100,000 records) | | | |
| | | | |
| Description of Test | AS/400 ILE | AS/400 PASE | % chg |
| | Optimized | Optimized | |
| | level 40, inline | + inline | |
| | | | |
| Random CLI Read/Update | | | |
| - Index Load (seq read) sec | 12 | 13 | 8.3 |
| - Random Read/Upd (50%/50%) | 371 | 386 | 4.0 |
| Total | 383 | 399 | 4.2 |

*Table 21.2 Comparison of DB2 CLI performance for AS/400 PASE and ILE*

The index load consisted reading sequentially through the file and building an array of 100,000 keys, with each key being 8 bytes long. The array of keys was then randomly accessed, and the random key value was then used to read the database table using a prebuilt index, alternating between a random read and a random read with update.

In AS/400 PASE test, four EBCDIC fields containing 30 characters of data were translated to ASCII. It is possible to turn off conversion by tagging the data in the database table with a native CCSID (coded character set identifier) of 65535. Conversion often happens even in an EBCDIC environment, for example when using national language support (NLS).

Since AS/400 PASE CLI support is calling under the covers the native CLI support which is written in ILE, the results show how AS/400 PASE call overhead discussed in the previous section factored in to this test. The overall difference of only 4.2% at the application level is due to the relatively large amount of time spent on the ILE side compared to the aggregate cost of the thousands of ILE calls made by the test. And the 4.2% difference also includes some additional processing in the AS/400 PASE CLI shared library to ensure thread safeness.

In the above tests CPU utilization, although not shown, was found to be proportional to response time.

**Commercial Application Ported to AS/400 PASE**

During the development of AS/400 PASE, a number of commercial applications were ported to AS/400 PASE to be used for testing. One of these applications was analyzed for performance. This application is characterized by being CPU intensive, with minimal database access, and uses stream I/O.

The application was tested for three different industries with various size companies being emulated. The results were compared to those obtained running a well tuned RS/6000 running AIX 4.3.2.

| | AIX | PASE | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Processor MHz:** | 340 | 400 | | | | | | |
| **Software** | AIX 4.3.2 | V4R5 | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| **Industry #1** | | | | | | | | |
| | **Elapsed Time (seconds)** | | | | | **Memory Used (KB)** | | |
| | | | | | | | | |
| **Input size** | **AIX** | **PASE Raw** | **PASE Scaled** | **PASE % of AIX** | | **AIX** | **PASE** | **PASE % of AIX** |
| 10,000 | 1,602 | 1,685 | 1,982 | 124% | | 530,044 | 586,288 | 111% |
| 100,000 | 27,819 | 29,528 | 34,739 | 125% | | 1,526,436 | 1,864,528 | 122% |
| 400,000 | 23,922 | 24,015 | 28,253 | 118% | | 2,106,260 | 2,064,496 | 98% |
| | | | | | | | | |
| | | | | | | | | |
| **Industry #2** | | | | | | | | |
| | **Elapsed Time (seconds)** | | | | | **Memory Used (KB)** | | |
| | | | | | | | | |
| **Input size** | **AIX** | **PASE Raw** | **PASE Scaled** | **PASE % of AIX** | | **AIX** | **PASE** | **PASE % of AIX** |
| 10,000 | 55 | 46 | 54 | 98% | | 240,692 | 33,712 | 14% |
| 150,000 | 437 | 383 | 451 | 103% | | 240,692 | 223,140 | 93% |
| 500,000 | 1,504 | 1,319 | 1,551 | 103% | | 711,756 | 684,368 | 96% |
| | | | | | | | | |
| | | | | | | | | |
| **Industry #3** | | | | | | | | |
| | **Elapsed Time (seconds)** | | | | | **Memory Used (KB)** | | |
| | | | | | | | | |
| **Input size** | **AIX** | **PASE Raw** | **PASE Scaled** | **PASE % of AIX** | | **AIX** | **PASE** | **PASE % of AIX** |
| 500,000 | 2,464 | 2,084 | 2,452 | 100% | | 544,396 | 527,312 | 97% |
| 750,000 | 4,914 | 3,102 | 3,649 | 74% | | 770,532 | 756,624 | 98% |
| 1,000,000 | 6,514 | 4,290 | 5,047 | 77% | | 997,740 | 980,784 | 98% |

*Table 21.3 Commercial Performance Example comparing AS/400 PASE and AIX*

The raw data was scaled to compensate for the differences in processor cycle time.

Performance varied depending on the nature type of data and the structure of the data, as well as the size of the dataset.

In general, the size of the dataset had minimal impact, and as can be seen, the performance difference varied by plus/minus 25%, leading to a conclusion, that performance on average is equivalent, but varies depending on the nature of the business and the structure of the data.

Memory utilization was also found to be similar for both AS/400 PASE and AIX.

## 21.3 V5R1 to V4R5 Release-to-Release Validation Workloads

The following three workloads, previously run on V4R5, were chosen to re-run on V5R1 to obtain a representative view of the performance comparison between the two releases of OS/400 PASE:

1. DB CLI
2. Netperf
3. i2 benchmarks

To ensure accurate comparsion data between the two releases, these workloads were run in both releases on the same physical iSeries hardware (same memory, processor speed and configuration, etc...).

### 21.3.1 DB CLI workload comparison

OS/400 PASE provides a shared library that enables access to the native iSeries DB2 Universal Database (UDB) Call Level Interface (CLI).

The DB CLI workload consists of reading sequentially through a file to build an array of 100,000 keys (index load). The array of keys is then randomly accessed, and the resulting key value is used to identify the database record to read and/or update.

**V5R1 vs V4R5 DB CLI Performance Measurement**

| | ILE % difference | PASE % difference |
|---|---|---|
| Total (secs) | -29.65 | -27.16 |

Note: The negative numbers represent the improvement in performance.

The table above shows that the total response time of DB CLI in both environments improve significantly in V5R1.

### 21.3.2 NetPerf Performance

This workload is a primitive-level function workload used to explore communication performance.

**V5R1 vs V4R5 Netperf Measurements**

| | transactions/sec | | Instructions | | CLT Cycles/transaction | | | SVR Cycles/transaction | |
|---|---|---|---|---|---|---|---|---|---|
| No. of Sessions | | | | | | | | | |
| 1 | 1.68% | | -2.78% | | 2.64% | | | -7.40% | |
| 8 | 9.21% | | -9.73% | | -9.62% | | | -6.35% | |
| 16 | 0.42% | | -8.35% | | -10.04% | | | -4.85% | |
| 96 | 2.47% | | -15.04% | | -6.74% | | | -2.69% | |
| 144 | 4.25% | | -15.59% | | -4.14% | | | 1.85% | |

Note: The negative number represents the improvement in performance.

The table of comparison above shows that we do not have any issues regarding the degradation of running NetPerf in V5R1 OS/400 PASE compared to V4R5.

### 21.3.3 i2 Performance

There are two i2 benchmarks (SCP and FP) which are used to explore the performance of commercial applications running in iSeries PASE.

**SCP and FP Benchmarks Performance Measurements**

| SCP Benchmark | | | |
|---|---|---|---|
| | **V4R5** | **V5R1** | % Difference |
| **Elapsed Time** | 31,347 | 29781 | -5.00% |

| FP Benchmark | | | |
|---|---|---|---|
| | **V4R5** | **V5R1** | % Difference |
| **Elapsed Time** | 4547 | 4277 | -6.31% |

Note: The negative numbers represent the improvement in performance.

The tables of comparison above show us that we have better performance running the i2 benchmarks in V5R1 OS/400 PASE.

### 21.3.4 Summary

The performance comparisons of the three workloads (DB CLI, NetPerf and i2 benchmarks) that represent in the sections above show that we significantly improve the performance of OS/400 PASE in V5R1 compared to V4R5.

# Chapter 22. IBM Workload Estimator for iSeries 400

## 22.1 Introduction

The purpose of the IBM Workload Estimator for iSeries 400 is to provide a comprehensive iSeries and AS/400 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. Recommendations will be for currently orderable system models.

The estimator is designed to be easy to use, typically with less than dozen questions per workload application and defaults for most workload questions and system assumptions based on common field experiences. The estimator can also be easily used with Performance Management/400 (PM/400) collected data.

The estimator can be used repeatedly to try "What-if" sizings. It has Growth estimation capabilities, inputs can be saved and restored later for reuse, and results along with the inputs can be printed in addition to being displayed.

## 22.2 Merging PM/400 data into the Workload Estimator

If you are using PM/400, you can easily merge your existing PM/400 data into the Estimator as a PM/400 workload. You can then size upgrades to your existing system based on your current workload or modify your current workload for future projections. This helps plan for future system requirements based on your existing utilization data. Multiple PM/400 system workloads can be merged in the Workload Estimator to size a server consolidation. The other workloads that Workload Estimator already sizes (i.e. Domino, Java, WebSphere, etc.) can also be combined with the PM/400 workload to size the needed upgrade.

The PM/400 data is easily merged into the Estimator while viewing your PM/400 graphs on the web. To view your PM/400 graphs on the web, go to http://www.ibm.com/eserver/iseries/pm400. Choose the 'click here to view your Management Summary Graph' button.

Follow these instructions to merge the PM/400 data into the Estimator:

1. Choose the 'Size my next upgrade' button.
2. Enter your PM/400 user id/password, if you have not already been required to do so. (If you've forgotten your password, back up one screen and choose the 'Resend Password' button.
3. Choose the 'IBM Workload Estimator for iSeries 400 or AS/400e' button.

Your PM/400 data is then passed into the Estimator. If this is your first time using PM/400 data with the Estimator, it is recommended that you take a few minutes to read the PM/400 integration with Workload Estimator help text. This text is found within the Estimator by clicking on Tutorials, followed by PM/400 Integration.

If you are not familiar with PM/400, refer to the iSeries Information Center at http://publib.boulder.ibm.com/pubs/html/as400/v5r1/ic2924/index.htm, search for PM/400, or refer to the PM/400 web page at http://www.ibm.com/eserver/iseries/pm400 for activation instructions.

## 22.3 Estimator Access

The IBM Workload Estimator for iSeries 400 is available to everyone via the Internet at http://as400service.ibm.com/estimator . Customers using the estimator should review all results with IBM or IBM Business Partner representatives.

New releases of the IBM Workload Estimator for iSeries 400 are planned 3 to 4 times per year. Since it's introduction, new releases have occurred every 3 to 4 months and have contained changes to add or update workloads, improve usability, add tool features, and add/update AS/400 and iSeries system hardware models.

As a web delivered tool, problems identified can be fixed and delivered independent of planned release schedules. As soon as problems are identified and fixed they are activated in the online tool.

Although use of the online version of the IBM Workload Estimator for iSeries 400 is strongly encouraged, IBM and IBM Business Partner representatives may also download a version of the estimator for running stand alone.

- The setup of a stand alone version of the Estimator requires downloading the estimator files plus several additional web server and Java components. Please note that due to the wide variance in Client hardware and Client OS (Windows) versions, there exists a great potential for problems and conflicts arising during the stand alone installation which cannot be anticipated in advance.
- For stand alone operation, the user is responsible to monitor the online version for updates and download updates otherwise risking running with outdated sizing algorithms and/or unfixed problems.
- The stand alone version of the Estimator can be found at http://as400service.ibm.com/supporthome.nsf/Document/16533356

## 22.4 Using the Estimator

To use the Estimator, you select one or more workloads and answer a few questions about each workload. Based on your answers, the Workload Estimator generates a system recommendation.

The general Flow of using the Estimator is as follows:

1. Open a Web browser to url http://as400service.ibm.com/estimator
   - Once each day you will be shown the license agreement screen and required to accept it
2. Next you will see a "Workload Selection" screen
   - Indicate whether including the workload for Existing (iSeries or AS/400) systems)
   - For "Type of Workload" use pulldown to select workload types (e.g. Existing system, Domino, Websphere, etc.). More than one workload may be included. Multiples of the same workload type are allowed, and by using the "Allow 5 more workloads" button additional workload entries can be added.
   - Customize the workload name if desired
   - Use the forward arrow (provided within the tool) to advance to the workload specific questions
3. Answer the questions for each workload screen presented.
   - Many questions are provided with defaults based on common lab and field experiences
   - The more the Estimator user understands about the environment being sized, and provides the information through answering the questions, the more representative the estimate will be.

    - Use the forward arrow to advance through each workload screen, and eventually to the “Selected System” results screen
4. The “Selected System” screen will show a recommended system model based on the resources required to support the workloads defined.
    - Options are provided to manually select other (typically larger) system models capable of supporting the workloads defined and/or change Estimator “Operational Assumptions” for performing the sizing and system selection
5. From this point one may choose to Save the Estimation, Print the results with all inputs included, project a system growth, or return to earlier screens and make modifications

Special features provided in the Estimator include:

- Extensive help text and links to performance white papers
    - Each screen displayed contains a “Help Menu” dropdown list with help pages for every different screen and workload type within the Estimator
    - Terms used in questions and fields contain links directly into the help text defining the term
    - Most terms also have fly over help text as quick reminders of the terms definition
- Operational Assumptions to control how the Estimator makes recommendations
    - The assumptions may be used once or stored as the defaults for the current Estimator user
    - Assumptions allow for specifying utilization thresholds, Disk protection, and system families
- Save (and Restore) of multiple estimations allowing for:
    - Continuation of estimations at a later time or
    - Reuse of earlier estimations as starting point of new estimations
- Recommendation based on Growth expectation
    - The Estimator user may enter a growth percentage and the Estimator will project additional resource requirements and make a system recommendation based on the need for growth.

---

## 22.5 What the Estimator is Not

The IBM Workload Estimator for iSeries 400 is provided as a sales aid.

- It does not perform Capacity Planning. BEST/1 is a capacity planner available for the iSeries 400. A Capacity Planner:
    - Performs extensive analytic or simulation modeling
    - Often models detailed transaction level interactions
    - Uses detailed performance data collected from an actual customer system
    - Projects transaction response times
- It does not perform system configuration. A system configurator:
    - Performs extensive hardware option and combinatorial checking
    - Accounts for all features, cards, racks, etc. of a complete system (order)
    - May provide total system pricing information
    - Does not provide performance sizing projections
- There are many workloads that do not presently exist in the IBM Workload Estimator for iSeries 400.
    - Some may be added in future releases as performance sizing information is understood
    - Some my not be added due to inability to represent workload performance parameters useable by the intended audience (i.e. not requiring a Subject Matter Expert to answer the questions)
    - Some (like many ERPs) have much more accurate sizing vehicles (i.e. ERP Competency Centers) and are best not to attempt to duplicate

## 22.6 Tips

- It is sometimes desirable to do multiple estimations with the Workload Estimator. One way to accomplish this is by returning to the Workload Selection screen and adding or removing workloads. This will keep the operational assumptions you chose to 'use this time', as well as any workloads that you did not remove from the first estimation. You could also re-invoke the estimator URL to start a new session of the Workload Estimator.
- The IBM Workload Estimator for iSeries 400 can only estimate the system resources and is limited by the information obtained through the questions. The Estimator recommendations are only part of the complete picture. Customer environment information not available to the Estimator and IBM sales and Business partner experience should be added to arrive at a final solution.

## 22.7 Summary

This section was designed to give a general overview of what the IBM Workload Estimator for iSeries 400 provides, where it can be accessed, how it can be used with PM/400 data, and what the Estimator is not. Because the Estimator is provided via the web and can change several times between issues of this Guide, the reader is encouraged to invoke the Estimator and read the extensive help provided with the currently active IBM Workload Estimator for iSeries 400.

Sizing recommendations start with benchmarks and performance measurements based on well-defined, consistent workloads. We have done many measurements and benchmarks for the iSeries and AS/400, and we are continuing to do them. However, we also want to provide rules of thumb for relating these performance measurements to other workloads that don't match the typical measured workload. We've used our experience running a large number of users in the Rochester development facility, along with feedback from customers and Business Partners who have ported their applications to develop these rules of thumb. Keep in mind, however, that many of these technologies are still new. Many customers are currently ramping up their production applications. We'll continue to refine these sizing recommendations 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 is as an estimate. This is particularly true with a robust system like iSeries and AS/400 that offers so many different capabilities where each installation will have unique performance characteristics and demands. The typical disclaimers that go with any performance estimate ("your experience might vary...") are especially true. We provide these sizing estimates as general guidelines, but can't guarantee their accuracy in all circumstances.

# Appendix A. CPW and CIW Descriptions

"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 iSeries and AS/400 systems, IBM has provided a measure called CPW to represent the relative computing power 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.

Over time, IBM analysts have identified two sets of characteristics the appear to represent a large number of environments on iSeries and AS/400 systems. Many applications tend to follow the same patterns as CPW - which stands for **Commercial Processing Workload**. These applications tend to have many jobs running brief transactions in an environment that is dominated by IBM system code performing database operations. Other applications tend to follow the same patterns as CIW - which stands for **Compute Intensive Workload**. These applications tend to have somewhat fewer jobs running transactions which spend a substantial amount of time in the application, itself. The term "Compute Intensive" does not mean that commercial processing is not done. It simply means that more CPU power is typically expended in each transaction because more work is done at the application level instead of at the IBM licensed internal code level.

## A.1 Commercial Processing Workload - CPW

The CPW rating of a system is generated using measurements of a specific workload that is maintained internally within the iSeries Systems Performance group. CPW is designed to evaluate a computer system and associated software in the commercial environment. It is rigidly defined for function, performance metrics, and price/performance metrics. It is NOT representative of any specific environment, but it is generally applicable to the commercial computing environment.

- What CPW is
  - Test of a range of data base applications, including simple and medium complexity updates, simple and medium complexity inquiries, realistic user interfaces, and a combination of interactive and batch activities.
  - Test of commitment control
  - Test of concurrent data access by large numbers of users running a single group of programs.
  - Reasonable approximation of a steady-state, data base oriented commercial application.

- What CPW is not:
  - An indication of the performance capabilities of a system for any specific customer situation
  - A test of "ad-hoc" (query) data base performance

- When to use CPW data
  - Approximate product positioning between different AS/400 models where the primary application is expected to be oriented to traditional commercial business uses (order entry, payroll, billing, etc.) using commitment control

## CPW Application Description

The CPW application simulates the database server of an online transaction processing (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.

Specific choices were made in creating CPW to try to best represent the relative positioning of iSeries and AS/400 systems. Some of the differences between CPW 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 CPW 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 workload
- Public benchmarks are super-tuned to obtain the best possible results for that specific benchmark, whereas for CPW 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 CPW 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 tend to stress extreme levels of scaling at very high CPU utilizations for very limited applications. To avoid misrepresenting the capacity of larger systems, CPW is measured at approximately 70% CPU utilization.
- 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, CPW is driven by a batch driver that is included as a part of the overall workload.

The net result is an application that IBM believes provides an excellent indicator of transaction processing performance capacity when comparing between members of the iSeries and AS/400 families. As indicated above, CPW is not intended to be a guarantee of performance, but can be viewed as a good indicator.

The CPW application simulates the database server of an online transaction processing (OLTP) environment. 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 retrival, 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.2 Compute Intensive Workload - CIW

Unlike CPW values, CIW values are not derived from specific measurements of a single workload. They are modeled projections which are based upon the characteristics of internal workloads such as Domino workloads and application server environments such as can be found with SAP or JDEdwards applications. CIW is meant to depict a workload that has the following characteristics:

- The majority of the system procession time is spent in the user (or software supplier) application instead of system services. For example, a Domino Mail and Calendar workload might spend 80% of the total processing time outside of OS/400, while the CPW workload spends most of its time in OS/400 database code.
- Compute intensive applications tend to be considerably less I/O intensive than most commercial application processing. That is, more time is spent manipulating each piece of data than in a CPW-like environment.

- What CIW is
  - ❖ Indicator of relative performance in environments where a significant amount of transaction time is spent computing within the processor
  - ❖ Indicator of some of the differences between this type of workload and a "commercial" workload

- What CIW is not:
  - ❖ An indication of the performance capabilities of a system for any specific customer situation
  - ❖ A measure of pure numeric-intensive computing

- When to use CIW data
  - ❖ Approximate product positioning between different iSeries or AS/400 models where the primary application spends much of its time in the application code or middleware.

**What guidelines exist to help decide whether my workload is CIW-like or CPW-like?**

An absolute assignment of a workload is difficult without doing a very detailed analysis. The general rules listed here are probable placements, but not absolute guarantees. The importance of having the two measures is to show that different workloads react differently to changes in the configuration. IBM's Workload Estimator tries to take some of these differences into account when projecting how a workload will fit into a new system (see Appendix B.)

In general, if your application is online transaction processing (order entry, billing, accounts receivable, and the like), it will be CPW-like. If there are many, many jobs that spend more time waiting for a user to enter data than for the system to process it, it is likely to be CPW-like. If a significant part of the transaction response time is spent in disk and communications I/O, it is likely to be CPW-like. If the primary purpose of the application is to retrieve, process, and store database information, it is likely to be CPW-like.

CIW-like workloads tend to process less data with more instructions than CPW-like workloads. If your application is an "information manipulator" rather than an "information processor", it is probable that it will be CIW-like. This includes web-servers where much time is spent in generating and sending web frames and pages. It also includes application servers, where data is received from end-users, massaged and formatted into transaction requests, and then sent on to another system to actually service the database requests. If an application is both a "manipulator" and a "processor", experience has shown that enough time is spent in the manipulation portion of the application that it tends to be the dominant factor and the

workload tends to be CIW-like. This is especially true of applications that are written using "modern" tools like Java, Websphere Application Server, and Websphere Commerce Suite. Another category that often fits into the CIW-like classification is overnight batch. Even though batch jobs often process a great deal of database work, there are relatively few jobs which means there is little switching of jobs from processor to processor. As a result, overnight batch data processing jobs sometimes act more like compute-intensive jobs.

**What are the differences in how these workloads react to hardware configurations?**

When you upgrade your system, the effectiveness of the upgrade may be affected by the type of workload you are running. CPW-like workloads tend to respond well to upgrades in memory and to processor upgrades where the increase in MHz of the processor is accompanied by improvements in the processor cache and memory subsystem. CIW-like workloads tend to respond more to pure MHz improvements and to increasing the number of processors. You may experience both kinds of improvements. For example, there may be a difference between the way the day-time OLTP application reacts to an upgrade and the way the night-time batch application reacts.

In a **CPW**-type workload, a lot of data is moved around and a wide variety of instructions are executed to manage the data. Because the transactions tend to be fairly short and because tasks are often waiting for new data to be brought from disk, processors are switched rapidly from task to task to task. This type of workload runs most efficiently when large amounts of the data it must process are readily available. Thus, it reacts favorably to large memory and large processor caches. We say that this type of workload is **cache-sensitive**. The bigger and faster the cache is, the more efficiently the workload runs (Note that cache is not an orderable feature. For iSeries, we attempt to balance processor upgrades with cache and memory subsystem upgrades whenever possible.) Increasing the MHz of the processor also helps, but you should not expect performance to scale directly with MHz unless other aspects of the system are equally improved. An example of this scenario can be found in V4R1, when the Model 640 systems were introduced as an upgrade path to Model 530 systems. The Model 640 systems actually had a lower MHz than the Model 530s, yet because they had much more cache and a much stronger memory implementation, they delivered a significantly higher CPW rating.

In a **CIW**-type workload, the situation is somewhat different. Compute intensive workloads tend to processes less data with more instructions. As a consequence, the opportunity for both instruction and data cache hits is much higher in this kind of workload. Furthermore, because the instruction path length tends to be longer, it is likely that processors will switch from task to task much less often. Having some cache is very important for these workloads, but having a big cache is not nearly as important as it is for CPW-like workloads. For systems that are designed with enough cache and memory to accommodate CPW-like work, there is usually more than enough to assist CIW-like work and so an increase in MHz will tend to have a more dramatic effect on these workloads than it does on CPW-like work. CIW-like workloads tend to be **MHz-sensitive**. Furthermore, since tasks stay resident on individual processors longer, we tend to see better scaling on multiprocessor systems.

CPW and CIW ratings for iSeries systems can be found in Appendix D of this manual.

# Appendix B. iSeries and AS/400 Sizing

In this section three sizing tools are referenced.

This section addresses:

- IBM Workload Estimator for iSeries 400 (formerly know as IBM Workload Estimator for AS/400 first Available 8/03/99)

  The purpose of the IBM Workload Estimator for iSeries 400 is to provide a comprehensive iSeries 400 and AS/400 sizing tool for new and existing customers interested in deploying new emerging workloads standalone or in combination with their current workloads. See Chapter 22 for a discussion on the IBM Workload Estimator for iSeries 400.

- the AS/400 Capacity Planner (BEST/1 for the AS/400)

  Best for MES upgrade sizing, or complex 'new business' system sizing.

- the AS/400 BATCH400 tool,

  Best for MES upgrade sizing where the 'Batch Window' is important.

## B.1 BEST/1 Capacity Planner for the AS/400

BEST/1 for the AS/400 is the product of an alliance with BMC Software, and is a part of the IBM Performance Tools/400 product. The capacity planner gives predicted performance information for response times, throughputs, and device utilizations based on estimated and/or measured workloads with a system configuration.

**Note:** *The BEST/1 for the AS/400 capacity planning tool is being withdrawn from the Performance Tools Licensed Product, effective with V5R2. No additional enhancements, including hardware table PTFs, will be available after V5R1. Current customers of BEST/1 should consider alternative capacity planning tools. One possible alternative IBM tool isPerformance Management/400 (PM/400) integrated with the IBM Workload Estimator for iSeries. While technically more of a "sizing" tool than a capacity planning tool, it does provide recommendations for upgrades from any iSeries or AS/400e model to the appropriate iSeries model and the user can adjust growth rates. The latest version utilizes customer performance trending data from PM/400. Other alternative capacity planning products are available from other vendors. More information on alternatives to BEST/1 may be available at a later time after V5R1 announce.*

### What It Does

The capacity planner helps to analyze the present and future performance requirements for iSeries and AS/400 systems. The capacity planner allows the use of predefined profiles and/or measured data to create an environment similar to the application environment required.

Use the predefined profiles for an initial proposal. Use the measurement capability alone if the current activity is growing or being analyzed. Mix the predefined profiles with the measured data if new applications are being added or the current ones are being changed significantly. The workloads are then mixed based on the number of local and/or remote devices specified. Optionally, the user can specify a response time or throughput objective for each of the workloads. These objectives (maximum for response time and minimum for throughput) represent the performance requirements.

After the workload has been defined, the capacity planner uses the measured configuration or allows the user to select from an IBM supplied list of configurations. The configuration and workload are analyzed and modeled to predict performance parameters such as response times, throughputs, and device utilizations. When measured configurations are not available, BEST/1 models perspective hardware configurations based on service times measured from a RAMP-C environment.

The capacity planner's evaluator then compares these numbers against a set of utilization guidelines and the optional response times or throughput objectives. If either the guidelines or objectives are not met, the evaluator recommends an upgrade to the system and reevaluates the adjusted system. This iterative process continues until a configuration is found that satisfies the guidelines and objectives.

Additionally, the planner includes a system growth function. The growth function allows the user to specify an anticipated growth rate over the entire system or by specific workloads. The capacity planner then estimates what configuration changes are required to sustain performance over time.

**What Is Supported**

The actual iSeries or AS/400 workload can be measured using the AS/400 Collection Services. BEST/1 uses this data to model system activities and provide workload support for the normal environment, and also functions such as PC Support (Work Station Feature and Shared Folders) and Display Station Passthrough (Source and Target).

BEST/1 also includes a set of predefined workloads which can be used to represent applications and workloads which are not measured. The predefined profiles include RAMP-C, Officevision/400, RTW, Batch, Spool, and others.

Support is provided for the various system functions, including checksum protection, purge option, and disk mirroring. In addition, BEST/1 also supports multiple memory pools, multiple priorities, multiple ASPs, batch, batch and interactive relationships, and the ability to model hardware enhancements the day they are announced.

BEST/1 for the AS/400 allows batch job analysis to be based on pool, priority, pathlength, and I/O characteristics the same as can be done for interactive jobs. This allows the user to set objectives for batch throughput, independent of interactive work, or in relation to interactive work.

BEST/1 provides a rated throughput for batch expressed in transactions per hour. This information can be used to estimate changes in throughput based on configuration or workload changes. BEST/1 does not provide detailed batch window analysis or job scheduling analysis. For modeling help in this area, reference Appendix B.2, “BATCH400.

The capacity planner can also be used to assist the System/36 customer in selecting an appropriately sized IBM iSeries or AS/400 system to meet their performance requirements. The capacity planner works with a System/36 migration utility procedure which is part of the System/36 release 5.1 coexistence PTF package (DK3700 or later) and System/36 Release 6. The utility uses the S/36 measured performance data created by the System Measurement Facility (SMF), and through modeling, translates the data into System/36 environment performance data for the Capacity Planner. The capacity planner can then be used to determine a system configuration that meets the anticipated performance needs.

**Where to Get It**

This capacity planner is part of the previously mentioned IBM AS/400 Performance Tools package which is a licensed program for the iSeries and AS/400 system (5763-PT1 in V3R1 and V3R2, 5716-PT1 in V3R6 and V3R7, 5769-PT1 in V4Rx, and 5722-PT1 in V5R1). This package also includes the measurement facilities needed to use the measurement interface capabilities of the capacity planner. This product's users guide includes more details on the measurements and capacity planner as well as specifics for the Capacity Planner System/36 Migration Utility option (*IBM AS/400 Programming: Performance Tools Guide*, SC41-8084 and *IBM AS/400 BEST/1 Capacity Planning Tool Guide*, SC41-5341). Other references also include the Capacity Planning Redbook (GG24-3908), and the BEST/1 Educational Video package (SK2T-6740-00 for 1/2" open reel, SK2T-6741-00 for 1/2" cartridge, SK2T-6742-00 for 1/4" cartridge, and SK2T-6743-00 for 8mm data tapes).

A Skill Dynamics Education course is available. The course is number CEM4930C - "AS/400 Capacity Planning using BEST/1."

**Announcement PTF summaries for BEST/1**

For the latest PTFs that are available to support the AS/400 Advanced Series with PowerPC AS technology, refer to HONE item RTA000089352. If you do not have access to HONE, please contact an IBM representative for this information.

**V3R6 Enhancements and PTF Changes for V3R1, V3R0.5, and V2R3**

- ***Analysis Enhancements***

  With the October Announcement, BEST/1 now supports the entire range of PowerPC-based Advanced Series AS/400 models.

  Specifically this includes the following systems:

  510 - 2144, 2143
  500 - 2142, 2141, 2140
  400 - 2133, 2132, 2131, 2130

  50S - 2121, 2120
  40S - 2110

  530 - 2153, 2152, 2151, 2150
  53S - 2156, 2155, 2154

For V3R1, two new "family" keywords have been added to the CPU Model Definition:

- *POWERAS - 510, 500, and 400 models
- *POWERSRV - 50S and 40S models

In V3R1, the "Upgrade to family" keyword for all CISC-based processors has been intentionally left as *ADVSYS or *ADVSRV. Explicit action is required by the user to configure BEST/1 to automatically trigger upgrade recommendations to PowerPC AS systems by changing the "Upgrade to family" keyword to *POWERAS or *POWERSRV on the processors where this is desired.

In V2R3, there is no "Upgrade to family" setting. BEST/1 only upgrades to CPU models which are currently available. As long as Advanced Series and Advanced Servers have their "Currently available" keyword set to Y=Yes, other CPU models will upgrade to them, because they appear earlier in the Hardware characteristics menu than the PowerPC AS models (with the exception of the 30S 2411 and 2412 models). If one wants BEST/1 to automatically trigger upgrade recommendations to PowerPC AS systems, one must change the "Currently available" keyword of the Advanced Series and/or Advanced Server models to N=No and change the PowerPC AS models to Y=Yes. V2R3 also contains internal rules which allow Server models to only upgrade to and from other Server models. Portables can not be upgraded to or from other CPU models.

BEST/1 now provides the capability to model the transition between CISC-based (Complex Instruction Set Computer) and RISC-based (Reduced Instruction Set Computer) or PowerPC technology based systems. IBM supplied values (not currently user-modifiable) are provided to manage the conversion at a transaction-level for the following values:

- CPU time
- Working Set Size
- I/O counts

These conversion factors are specified for General purpose, CPU intensive, I/O intensive, and Development workloads and are applied during analysis when BEST/1 models are analyzed with a configuration containing a RISC-based CPU model. The current conversion factors are listed in Table B.1 below:

*Table B.1. CISC-to-RISC Conversion Factors*

| Description | CPU Time | Working Set Size* | I/O Counts |
|---|---|---|---|
| General purpose workload type | 1.00 | 2.00 | 1.00 |
| CPU intensive workloads where the CPU per I/O rate is high | 0.50 | 1.50 | 1.00 |
| I/O intensive workloads where the CPU per I/O rate is low | 1.00 | 2.30 | 1.00 |
| Development workload where compile and debug are the bulk of the work | 1.50 | 5.00 | 1.00 |

Note:
(*) In addition to the working set size conversion factor, the minimum RISC machine pool requirement of 16 MB (regardless of total system mainstore memory size) is included when recommending pool sizes during a CISC-to-RISC upgrade. In extreme cases this will cause excess memory to be recommended during a growth analysis because BEST/1

| Description | CPU Time | Working Set Size* | I/O Counts |
|---|---|---|---|
| preserves the ratio of the machine pool to total mainstore memory as the other pools are increased in size due to workload growth.<br><br>DISCLAIMER: CONVERSION FACTORS AND BEST/1 RESULTS HAVE NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND ARE DISTRIBUTED ON AN "AS IS" BASIS AT THIS TIME WITHOUT ANY WARRANTY EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE. The use of BEST/1 results is a customer responsibility and is dependent on the customer's operational environment; customers applying BEST/1 results do so at their own risk. Conversion factors may change as additional experience is gained by analyzing comparable performance data on CISC and RISC systems. New PTFs will be made available as appropriate. | | | |

*NORMAL and *BATCHJOB are the default and recommended settings for workload type and, furthermore, are the only workload types available in V2R3 and V3R0.5. These types will refer to conversion factors which are expected to provide good results on most general purpose customer configurations. The *TRNxxxxx workload types represent specific workload characteristics, and are provided for users who understand how they match the workloads in their model per Table B.2 below:

*Table B.2. Workload Type Characteristics*

| Description | Normal | Batch |
|---|---|---|
| General purpose workload type | *NORMAL | *BATCHJOB |
| CPU intensive workloads where the CPU per I/O rate is high | *TRNNORM1 | *TRNBAT1 |
| I/O intensive workloads where the CPU per I/O rate is low | *TRNNORM2 | *TRNBAT2 |
| Development workload where compile and debug are the bulk of the work | *TRNNORM3 | *TRNBAT3 |

Additional description:

- *NORMAL and *BATCHJOB

  These default workloads are represented by traditional commercial applications. CPU profiles for these workloads have up to 10-20% of their CPU time spent in application programs and the remaining 80+% spent in operating system programs. This is because typical RPG and COBOL business applications utilize a significant amount of system services such as database I/O, query processing, workstation/printer processing, and communication I/O.

- *TRNNORM1 and *TRNBAT1

  These types of workloads are referred to as Application Compute Intensive. In these workloads, a majority of the CPU time is is spent in application programs. Although these applications can be written in other languages, they are typically written in in ILE C. Examples of these types of applications are: financial modeling applications which do a significant amount of numeric calculations, statistical analysis applications, 4GL interpreters, and applications which implement complex business rules.

- *TRNNORM2 and *TRNBAT2

  These types of workloads are characterized on a properly configured and tuned system by the majority of application time being spent doing I/O.

- *TRNNORM3 and *TRNBAT3

  These types of workloads are characterized by the OPM development environment where compiles are done with *NOPTIMIZE. Specifying *OPTIMIZE can increase the CPU time. For ILE development environments use the *NORMAL workload type.

To aid in the classification of a workload Table B.3 below illustrates some sample CPU per I/O values for various DASD response times. One can analyze their BEST/1 workload details to determine if any workload types should be changed. Calculate the CPU per I/O value by dividing the CPU seconds per transaction by the total reads and writes per transaction. For example, if the workload has a single function executing 1 function per user, the function defines a single transaction executing 1 transaction per function, and the transaction specifies 100 I/Os and 5 CPU seconds, then it is characterized by a CPU per I/O value of 0.05 (5 / 100 = 0.05). Thus if the average DASD response time for the configuration is 20 milliseconds this would indicate the *NORMAL classification is the correct setting for the workload.

(Note: if there is more than 1 transaction definition in the workload, one will need to account for the number of transactions in the calculation such that the CPU per I/O value is a weighted average: if 10 transactions have a value of 0.05 and 40 transactions have a value of 0.15, then the weighted average value is 0.13).

*Table B.3. Sample CPU per I/O Values and the Appropriate Workload Type*

| Avg I/O Resp Time | I/O Intensive *TRNNORM2 or *TRNBAT2 | *NORMAL or *BATCHJOB | CPU Intensive *TRNNORM1 or *TRNBAT1 |
|---|---|---|---|
| 10 msec | < 0.003 | 0.003 to 0.040 | > 0.040 |
| 20 msec | < 0.005 | 0.005 to 0.080 | > 0.080 |
| 30 msec | < 0.008 | 0.008 to 0.120 | > 0.120 |

- ***Performance Enhancements for V3R6***

  BEST/1 now runs in ILE instead of EPM, resulting in better performance.

- ***Model Creation Enhancements for V3R6***

  BEST/1 now supports the ADV36 Job Type.

- ***Usability Enhancements for V3R6***

  BEST/1 supports changing the feature of multiple IOP's, controllers and arms during a single operation.

- ***Predefined Workloads for V3R6***

  Additional server predefined workloads have been added to provide workloads for multimedia environments.

### V3R7 Enhancements and PTF Changes for V3R7 and V4R1

- ***Performance Enhancements for V3R7***

  IBM has optimized machine code for some processes for V3R7, as compared to V3R6, which results in improved performance. Consequently, the OS/400 release being run is now part of the BEST/1 model. In this way, you can model the effect of upgrading your release.

- ***Usability Enhancements for V3R7***

  ***BASIC User Level:** Inexperienced or irregular users of BEST/1 can focus on primary capacity planning activities through "hiding" advanced functionality. You can specify the user level using a new parameter on the STRBEST command. You can also switch between Basic and Advanced user levels on the Work with BEST/1 Model display.

### V4R2 Enhancements and PTF Changes for V4R2 and V4R3

- ***Usability Enhancements for V4R2***

  **Modeling of Input/Output Adapters (IOAs):** IOAs are represented explicitly in the BEST/1 configuration. This applies to both disk IOAs and comm IOAs. Only multifunction IOPs can be connected to IOAs. Utilizations and other performance results are included in the reports.

  **Measured IOP Utilizations:** Measured values for disk, multifunction, LAN, and WAN IOP utilizations are determined differently than in prior releases of BEST/1. Measured multifunction IOP utilization was included in both the disk category and the appropriate communications category prior to V4R2. In this release, measured multifunction IOP utilization is reported separately. As a result, when reading a pre-V4R2 model, the measured IOP utilizationsare calculated again, and may vary from pre-V4R2 results.

### V4R4 Enhancements

- ***Usability Enhancements for V4R4***

  **Modeling of DASD Compression:** The effects of ASP-based data compression are modeled explicitly. You can specify whether arms in an ASP have compression turned on or off.

  **DASD I/O Distribution:** I/O distribution within an ASP is based on the capacity of each disk arm. For example, if your ASP has a 1 GB arm and a 2 GB arm, the 2 GB arm will receive twice as many I/Os as the 1 GB arm.

  **Disk Utilization Guidelines at the ASP Level**: You can specify different disk utilization guidelines for each ASP in your modek, These guidelines are specified in each model on the Edit ASP display.

  **Support for 64 Main Storage Pools:** OS/400 for V4R3 increases the maximum number of storage pools from 16 to 64. BEST/1 for V4R4 supports this increased number.

**Support for Logical Partitioning:** BEST/1 models the performance of an individual logical partition, based on the number of processors, main storage, and various I/O devices which are assigned to that partition. Logical partitioning on the AS/400 is a new mode of machine operation where multiple copies of OS/400 run on a single AS/400. A logical partition is a collection of machine resources that are capable of running an operating system. Partitions operate independently and are isolated from each other logically.

**V4R5 Enhancements**

- ***Usability Enhancements for V4R5***

**Support for PCI Nodes:** Support for this new bus architecture is included in configuration checking, correcting, and recommendations. The number of available PCI slots is used when determining how many PCI-based IOPs can be added to a system.

**Emerging Workloads:** You can model differences in workload scaling across CPU models by specifying an application type at the transaction level. Each application type has a performance adjustment for each CPU model. This factor is applied during analysis, since a single CPU performance rating is no longer sufficient to indicate how workloads will scale across CPU models. Please note that the application type of each transaction can only be determined during the creation of a BEST/1 model if you have turned on the internal data parameter when starting the performance monitor, as in: STRPFRMON INTDTA(*YES).

**V5R1 Enhancements**

**No New Functional Enhancements:** No new functional enhancements have been made to the V5R1 version of BEST/1. The hardware table, however, has been updated to provide support for V5R1 announced hardware. The hardware table changes are available for V4R4 and V4R5 versions of BEST/1, as well as for V5R1.

---

## B.2 BATCH400

BATCH400 is a tool for Batch Window Analysis available for V3R6+ systems. It is an internal use only tool at this time. Instructions for requesting a copy are at the end of this description.

BATCH400 is a tool to enable I Series and AS/400 batch window analysis to be done using information collected by the OS/400 Collection Services.

BATCH400 addresses the often asked question: 'What can I do to my system in order to meet my overnight batch run-time requirements (also known as the Batch Window).'

The BATCH400 tool creates a 'model' from Collection Services performance data. This model will reside in a file called 'QSBSCHED' in the target library. The tool can then be asked to analyze the model and provide results for various 'what-if' conditions. Individual batch job run-time, and overall batch window run-times will be reported by this tool.

BATCH400 Output description:

1. Configuration summary shows the current and modeled hardware for DASD and CPU.

2. Job Statistics show the modeled result followed by the original (probably measured) data for each workload. Workloads are given short names (like b6) that represent either a single batch job or a collection of jobs grouped together. A listing at the end of the output shows the job number/user/name associated with each workload name. A short name like b6 indicates that a job is in the 6th 'thread' of jobs, since the letter is 'b', it is the second job in the thread (a6 being the first). Most other fields in this section are self explanatory. (Tr/Sec) is the Sync I/O per second rate for batch jobs. (Tr/Sec) is the Interactive transaction/second for the interactive workload. (Int Rt) is the Interactive response time. (Bat eff) is the batch efficiency of the workload, a value of 1.0 means the the workload is 100% CPU bound. (ExWait) is the time we had to add to the workload to account for the entire time the job was present in the AS/400, large values here can indicate either delay jobs, or excessive DB contention.

3. Thread summary shows the start/stop/elapsed time for entire threads.

4. Graph of Threads vs. Time of Day shows a 'horizontal' view of all threads in the model. This output is very handy in showing the relationship of job transitions within threads. It might indicate opportunities to break threads up to allow jobs to start earlier and run in parallel with jobs currently running in a sequential order.

5. Total CPU utilization shows a 'horizontal' view of how busy the CPU is. This report is on the same time-line as the previous Threads report.

6. Bar chart of Thread elapsed Times shows a comparison of all threads based on end-to-end thread run time. All threads start at time of zero and reach up into this report depending on how long they run. This report shows a 'vertical' depiction of each thread, and will occasionally show better job transition details than the 'horizontal' view noted above. This report can help to identify the longest running thread and possible candidate workloads to be improved, or scheduled for different times.

7. The rest of the output shows the model that is stored in QSBSCHED. This is the model that was used for the analysis. The config summary, and workload details are followed by a listing of the workload definitions. This workload definition usually shows:

    - interactive workloads which are a summary of all interactive jobs at a given priority level (type 1 workloads).

    - System workloads which are a summary of various VMC tasks and 'short running' batch jobs at a given priority level (type 3).

    - Batch workloads which are a summary of individual batch jobs which are usually the focus of the analysis (type 2).

    - Async workloads which are a summary of all the Asynchronous I/O tasks running on the system (type 4).

After looking at the results, use the BATCH400 *CHANGE option to invoke WRKOBJPDM. This will allow you to edit the model. You can alter the job dependencies. Maybe job b6 doesn't have to follow job

a6 in thread 6. You can remove the previous job linkage for b6 and rename b6 job to b12 (where 12 is the next available thread number). Upon saving and exiting, BATCH400 will analyze the new model.

For now this tool is on the IBM intranet at the following URL:

http://ca-web.rchland.ibm.com/perform/perftool/batch400/batch400.htm

## B.3 Performance Data Collection Services

Collecting performance data with Collection Services is an operating system function designed to run continuously that collects system and job level performance data at regular intervals which can be set from 15 seconds to 1 hour. It runs several collection routines called probes which collect data from many system resources including jobs, disk units, IOPs, buses, pools, and communication lines. Collection Services is the replacement for the Performance Monitor function which you may have used in previous releases to collect performance data by running the STRPFRMON command. Collection Services has been available in OS/400 since V4R4. The Performance Monitor has remained on the system through V4R5 to give you time to switch over to the new Collection Services function.

### Why the Performance Monitor was replaced

The Performance Monitor was designed for the System/38 at a time when a system with a couple hundred jobs was a large and fully utilized system. As the AS/400 and then iSeries continued to get bigger and faster, the Performance Monitor could no longer scale to handle the thousands of jobs and threads that has become common in a modern computing environment. The limitations inherent to the Performance Monitor design made it difficult to handle the increased workloads and faster system resources without severely degrading the performance of the system. Clearly, nothing could be worse than finding out that a tool used for diagnosing system performance problems was actually making the performance problem more severe. It was not uncommon for the Performance Monitor to use as much as 15% of the CPU when collecting performance data on a system that was running several thousand jobs. On systems where CPU had already spiked to near maximum capacity, it was unacceptable to run a tool that consumed that much CPU when attempting to diagnose the problem. Many enhancements were made to the Performance Monitor over the years to improve its efficiency and scalability, but it became apparent that significant improvements could not be made without a complete redesign of the function.

### Why Collection Services is more efficient

Collection Services is much more efficient than the Performance Monitor because it has a much improved method for storing the performance data that is collected. A system object called a management collection object (*MGTCOL) was created in V4R4 to store Collection Services data. The management collection object takes advantage of terraspace support to make it a much more efficient way to store large quantities of performance data. The Performance Monitor stored the data it collected in over 30 database files, but Collection Services stores the data in a single collection object and supports a release independent design which allows you to move data to a system at a different release without requiring database file conversions.

Since many of the reporting, analysis, and trending tools like Performance Tools/400, PM/400, and BEST/1 use the Performance Monitor database files, it was important to maintain the ability to generate

those files. A command called CRTPFRDTA (Create Performance Data) can be used to create the database files from the contents of the management collection object. The CRTPFRDTA command gives you the flexibility to generate only the database files you need to analyze a specific situation. If you decide that you always want to generate the database as the Performance Monitor did, you can configure Collection Services to run CRTPFRDTA as a low-priority batch job while data is being collected. Separating the collection of the data from the database generation, and running the database function at a lower priority are key reasons why Collection Services is much more efficient and can collect data from large quantities of jobs and threads at very frequent intervals. With Collection Services, you can collect performance data at intervals as frequent as every 15 seconds if you need that level of granularity to diagnose a performance problem. Collection Services also supports collection intervals of 30 seconds, and 1, 5, 15, 30, and 60 minutes.

The overhead associated with collecting performance data is now minimal enough that Collection Services can run continuously, no matter what workload is being run on your system. By contrast, some customers could only afford the overhead of the Performance Monitor for a couple hours at a time for a few periods per week. If Collection Services is run continuously as designed, you will capture the data needed to analyze and solve many performance slowdowns before they turn into a serious problem. Prior to Collection Services, if you encountered a performance problem during one of the large windows when the Performance Monitor was not running, you often did not have the data needed to understand what caused the problem.

**Starting Collection Services**

The Performance Monitor was started using the STRPFRMON command, or by using option 2 on the Performance menu (GO PERFORM). STRPFRMON no longer exists, but option 2 on the Performance menu now supports the Collection Services facility. You can also start Collection Services by using the Management Central GUI, or by using the Start Collector API. For more details on these options, see Performance Overview under the Systems Management topic in the V5R1 Information Center which is available at http://www.ibm.com/eserver/iseries/infocenter.

When using the Management Central GUI, you fill find that it gives you much more flexibility than the Performance Monitor to collect only the performance data you are interested in. Collection Services data is organized into over 20 categories and you have the ability to turn on and off each category or select a predefined profile containing commonly used categories. For example, if you do not have a need to monitor the performance of SNADS transaction data on a regular basis, you can choose to turn that category off so that SNADS transaction data is not collected.

An option existed on the STRPFRMON command to start the Performance Monitor in trace mode. This option was useful to identify lock contention problems in an application. Trace mode can still be used, but it was not integrated into the start options of Collection Services, since Collection Services is intended to be run continuously and trace mode is not. To run the same trace mode facility that was available through the Performance Monitor, you need to use two new commands called STRPFRTRC (Start Performance Trace) and ENDPFRTRC (End Performance Trace). For more information on these commands, see Performance Overview under the Systems Management topic in the V5R1 Information Center which is available at http://www.ibm.com/eserver/iseries/infocenter.

# Appendix C. DASD IOP/IOA Device Characteristics

This appendix describes the DASD models supported by the 6502, 6512,6530, 6532, 6751 and 6754 IOPs and the 2726, 2740, 2741, 2748, 2763, 2778, 4748, 4778, 9728 and 9767 IOAs.

### 6502/6512 Disk Unit Controller for RAID

Feature #6502 is a disk controller with a 2MB write-cache. Feature #6512 is an enhanced disk controller with a 4MB write-cache. Both can provide RAID-5 protection for up to 16 internal disk units installed in the Storage Expansion Units (#5051/5052). Additionally, disk units attached with #6502 or #6512 and not in a RAID array can be mirrored and/or unprotected. In the RAID configuration, disk unit protection is provided at less cost than mirroring, and with greater performance than system checksums. The 6502 and 6512 also supports mixing different internal disk features on the same controller. RAIDed disks must have the same capacity.

### 6502/6512 Supported DASD Models

*Table C.1. 6502/6512 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6605 | 050 | 1031 | No | Yes |
| 6605 | 070 | 1031 | Yes | Yes |
| 6605 | 072 | 902 | Yes | Yes |
| 6605 | 074 | 773 | Yes | Yes |
| 6606 | 050 | 1967 | No | Yes |
| 6606 | 070 | 1967 | Yes | Yes |
| 6606 | 072 | 1721 | Yes | Yes |
| 6606 | 074 | 1475 | Yes | Yes |
| 6607 | 050 | 4194 | No | Yes |
| 6607 | 070 | 4194 | Yes | Yes |
| 6607 | 072 | 3669 | Yes | Yes |
| 6607 | 074 | 3145 | Yes | Yes |
| 6713 | 050 | 8589 | No | Yes |
| 6713 | 070 | 8589 | Yes | Yes |
| 6713 | 072 | 7515 | Yes | Yes |
| 6713 | 074 | 6441 | Yes | Yes |
| 6714 | 050 | 17548 | No | Yes |
| 6714 | 070 | 17548 | Yes | Yes |
| 6714 | 072 | 15354 | Yes | Yes |
| 6714 | 074 | 13161 | Yes | Yes |

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of two arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives. 1 GB and larger disk units can be RAID-5 protected by the controller. Each System Unit Expansion Tower can support up to 16 disk units on 1 6502/6512 disk controller when the #5052 Storage Expansion Unit is installed. Each DASD Expansion Tower can support up to 32 disk units on 2 6502/6512 disk controllers.

For the Model 400, up to eight 1 GB disk units can be supported when using the #7117 Integrated Expansion Unit.

### 6530 Storage Device Controller

The #6530 is a Storage controller for up to 16 disk units installed in the Storage Expansion Units (#5051/5052). The 6530 does NOT have a Write Cache and does NOT support RAID. The 6530 also supports mixing different internal disk features on the same controller.

### 6530 Supported DASD Models

*Table C.2. 6530 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6605 | 030 | 1031 | No | No |
| 6606 | 030 | 1967 | No | No |
| 6607 | 030 | 4194 | No | No |
| 6713 | 030 | 8589 | No | No |
| 6714 | 030 | 17548 | No | No |

Each System Unit Expansion Tower can support up to 16 disk units on 1 6530 disk controller when the #5052 Storage Expansion Unit is installed. Each DASD Expansion Tower can support up to 32 disk units on 2 6530 disk controllers.

### 6533/6754 Disk Unit Controller for RAID

Feature #6533 (also #6532) is a disk controller with a 4MB write-cache. It can provide RAID-5 protection for up to 16 internal disk units installed in the Storage Expansion Units. Feature #6754 (also #6751) is a Multi-Function IOP with a 4MB write-cache. It can provide RAID-5 protection for up to 20 internal disk units installed in the System Unit. Additionally, disk units attached with the #6533 or #6754 and not in a RAID array can be mirrored and/or unprotected. The 6533 and 6754 also support mixing different internal disk features on the same controller. RAIDed disks must have the same capacity.

### 6533/6754 Supported DASD Models

*Table C.3. 6533/6754 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6606 | 050 | 1967 | No | Yes |
| 6806 | 050 | 1967 | No | Yes |
| 6606 | 070 | 1967 | Yes | Yes |
| 6806 | 070 | 1967 | Yes | Yes |
| 6606 | 072 | 1721 | Yes | Yes |
| 6806 | 072 | 1721 | Yes | Yes |
| 6606 | 074 | 1475 | Yes | Yes |
| 6806 | 074 | 1475 | Yes | Yes |
| 6607 | 050 | 4194 | No | Yes |
| 6807 | 050 | 4194 | No | Yes |
| 6607 | 070 | 4194 | Yes | Yes |
| 6807 | 070 | 4194 | Yes | Yes |
| 6607 | 072 | 3669 | Yes | Yes |
| 6807 | 072 | 3669 | Yes | Yes |
| 6607 | 074 | 3145 | Yes | Yes |
| 6807 | 074 | 3145 | Yes | Yes |
| 6713 | 050 | 8589 | No | Yes |
| 6813 | 050 | 8589 | No | Yes |
| 6713 | 070 | 8589 | Yes | Yes |
| 6813 | 070 | 8689 | Yes | Yes |
| 6713 | 072 | 7515 | Yes | Yes |
| 6813 | 072 | 7515 | Yes | Yes |
| 6713 | 074 | 6441 | Yes | Yes |
| 6813 | 074 | 6441 | Yes | Yes |
| 6717 | 050 | 8589 | No | Yes |
| 6717 | 070 | 8589 | Yes | Yes |
| 6717 | 072 | 7515 | Yes | Yes |
| 6717 | 074 | 6441 | Yes | Yes |
| 6714 | 050 | 17548 | No | Yes |
| 6714 | 070 | 14548 | Yes | Yes |
| 6714 | 072 | 15354 | Yes | Yes |
| 6714 | 074 | 13161 | Yes | Yes |

**Note**: 6806, 6807, 6813, and 6714 are Ultra-SCSI DASD and 6717 is a SCSI Wide-Ultra2 DASD.

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of two arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives. Each System Unit Expansion Tower can support up to 16 disk units on 1 6533 disk controller. Each DASD Expansion Tower can support up to 32 disk units on 2 6533 disk controllers.

For the Models 640 and 650, up to 20 disk units can be supported on 1 6754 MFIOP.

### 2741/2740/2726 Disk Unit Controller for RAID

Feature #2741 (also #2726) is a disk controller with a 4MB write-cache. It can provide RAID-5 protection for up to 15 internal disk units installed in the PCI System Unit, PCI Expansion Unit or PCI Expansion Tower. Feature #2740 is a low cost disk controller with a 4MB write-cache that is targeted for smaller systems. It can provide RAID-5 protection for up to 10 internal disk units installed in the PCI System Unit or PCI Expansion Unit. Additionally, disk units attached with the #2741 or #2740 and not in a RAID array

can be mirrored and/or unprotected. The 2741 and 2740 also support mixing different internal disk features on the same controller. RAIDed disks must have the same capacity.

### 2741/2740/2726 Supported DASD Models

*Table C.4. 2741/2740/2726 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6606 | 050 | 1967 | No | Yes |
| 6806 | 050 | 1967 | No | Yes |
| 6606 | 070 | 1967 | Yes | Yes |
| 6806 | 070 | 1967 | Yes | Yes |
| 6606 | 072 | 1721 | Yes | Yes |
| 6806 | 072 | 1721 | Yes | Yes |
| 6606 | 074 | 1475 | Yes | Yes |
| 6806 | 074 | 1475 | Yes | Yes |
| 6607 | 050 | 4194 | No | Yes |
| 6807 | 050 | 4194 | No | Yes |
| 6607 | 070 | 4194 | Yes | Yes |
| 6807 | 070 | 4194 | Yes | Yes |
| 6607 | 072 | 3669 | Yes | Yes |
| 6807 | 072 | 3669 | Yes | Yes |
| 6607 | 074 | 3145 | Yes | Yes |
| 6807 | 074 | 3145 | Yes | Yes |
| 6713 | 050 | 8589 | No | Yes |
| 6813 | 050 | 8589 | No | Yes |
| 6713 | 070 | 8689 | Yes | Yes |
| 6813 | 070 | 8589 | Yes | Yes |
| 6713 | 072 | 7515 | Yes | Yes |
| 6813 | 072 | 7515 | Yes | Yes |
| 6713 | 074 | 6441 | Yes | Yes |
| 6813 | 074 | 6441 | Yes | Yes |
| 6717 | 050 | 8589 | No | Yes |
| 6717 | 070 | 8589 | Yes | Yes |
| 6717 | 072 | 7515 | Yes | Yes |
| 6717 | 074 | 6441 | Yes | Yes |
| 6714 | 050 | 17548 | No | Yes |
| 6714 | 070 | 17548 | Yes | Yes |
| 6714 | 072 | 15354 | Yes | Yes |
| 6714 | 074 | 13161 | Yes | Yes |

**Note**: 6806, 6807, 6813, and 6714 are Ultra-SCSI DASD and 6717 is a SCSI Wide-Ultra2 DASD.

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of two arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives.

### 2748/2778 PCI RAID Unit Controller**s**

Feature #2748 is a PCI disk controller with a 26MB write-cache. Feature #2778 is a PCI disk controller with a 26MB compressed write-cache. Both provide RAID-5 protection for up to 15 internal disk units installed in the PCI System Unit, PCI Expansion Unit, or PCI Expansion Tower. Additionally, disk units attached with the #2748 or #2778 and not in a RAID array can be mirrored and/or unprotected. Both also support mixing different internal disk features on the same controller. RAIDed disks must have the same capacity. When DASD Compression is enabled, the write-cache is limited to 4MB.

## 2748/2778 Supported DASD Models

*Table C.5. 2748/2778 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6606 | 050 | 1967 | No | Yes |
| 6806 | 050 | 1967 | No | Yes |
| 6606 | 070 | 1967 | Yes | Yes |
| 6806 | 070 | 1967 | Yes | Yes |
| 6606 | 072 | 1721 | Yes | Yes |
| 6806 | 072 | 1721 | Yes | Yes |
| 6606 | 074 | 1475 | Yes | Yes |
| 6806 | 074 | 1475 | Yes | Yes |
| 6607 | 050 | 4194 | No | Yes |
| 6807 | 050 | 4194 | No | Yes |
| 6607 | 070 | 4194 | Yes | Yes |
| 6807 | 070 | 4194 | Yes | Yes |
| 6607 | 072 | 3669 | Yes | Yes |
| 6807 | 072 | 3669 | Yes | Yes |
| 6607 | 074 | 3145 | Yes | Yes |
| 6807 | 074 | 3145 | Yes | Yes |
| 6713 | 050 | 8589 | No | Yes |
| 6813 | 050 | 8589 | No | Yes |
| 6713 | 070 | 8689 | Yes | Yes |
| 6813 | 070 | 8589 | Yes | Yes |
| 6713 | 072 | 7515 | Yes | Yes |
| 6813 | 072 | 7515 | Yes | Yes |
| 6713 | 074 | 6441 | Yes | Yes |
| 6813 | 074 | 6441 | Yes | Yes |
| 6717 | 050 | 8589 | No | Yes |
| 6717 | 070 | 8589 | Yes | Yes |
| 6717 | 072 | 7515 | Yes | Yes |
| 6717 | 074 | 6441 | Yes | Yes |
| 6714 | 050 | 17548 | No | Yes |
| 6714 | 070 | 17548 | Yes | Yes |
| 6714 | 072 | 15354 | Yes | Yes |
| 6714 | 074 | 13161 | Yes | Yes |
| 6718 | 050 | 17548 | No | Yes |
| 6718 | 070 | 17548 | Yes | Yes |
| 6718 | 072 | 15354 | Yes | Yes |
| 6718 | 074 | 13161 | Yes | Yes |

**Note**: 6806, 6807, 6813, and 6714 are Ultra-SCSI DASD; 6717 and 6718 are SCSI Wide-Ultra2 DASD.

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of three arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives.

## 2763 PCI RAID Unit Controller

Feature #2763 is a PCI disk controller with a 10MB write-cache. It can provide RAID-5 protection for up to 6 internal disk units installed in the Model 270/820 PCI System Unit or PCI Expansion Tower and up to 12 internal disk units installed in the PCI System Unit Expansion. Additionally, disk units attached with the #2763 and not in a RAID array can be mirrored and/or unprotected. The 2763 also supports mixing different internal disk features on the same controller. RAIDed disks must have the same capacity.

### 2763 Supported DASD Models

*Table C.6. 2763 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 4314 | 050 | 8589 | No | Yes |
| 4314 | 070 | 8589 | Yes | Yes |
| 4314 | 072 | 7515 | Yes | Yes |
| 4314 | 074 | 6441 | Yes | Yes |
| 4317 | 050 | 8589 | No | Yes |
| 4317 | 070 | 8589 | Yes | Yes |
| 4317 | 072 | 7515 | Yes | Yes |
| 4317 | 074 | 6441 | Yes | Yes |
| 4318 | 050 | 17548 | No | Yes |
| 4318 | 070 | 17548 | Yes | Yes |
| 4318 | 072 | 15354 | Yes | Yes |
| 4318 | 074 | 13161 | Yes | Yes |
| 4324 | 050 | 17548 | No | Yes |
| 4324 | 070 | 17548 | Yes | Yes |
| 4324 | 072 | 15354 | Yes | Yes |
| 4324 | 074 | 13161 | Yes | Yes |

**Note**: 4314 and 4324 are Ultra-SCSI DASD; 4317 and 4318 are SCSI Wide-Ultra2 DASD.

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of three arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives.

### 4748/4778 PCI RAID Unit Controller

Feature #4748 is a PCI disk controller with a 26MB write-cache. Feature #4778 is a PCI disk controller with a 26MB compressed write-cache. Both can provide RAID-5 protection for up to 6 internal disk units installed in the Model 270/820 PCI System Unit or PCI Expansion Tower, up to 15 internal disk units installed in PCI I/O Towers and up to 18 internal disk units installed in the Model 270 PCI System Unit with the System Unit Expansion added. Additionally, disk units attached with the #4748 or #4778 and not in a RAID array can be mirrored and/or unprotected. Both also support mixing different internal disk features on the same controller. RAIDed disks must have the same capacity. When DASD Compression is enabled, the write-cache is limited to 4MB.

### 4748/4778 Supported DASD Models

*Table C.7. 4748/4778 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 4314 | 050 | 8589 | No | Yes |
| 4314 | 070 | 8589 | Yes | Yes |
| 4314 | 072 | 7515 | Yes | Yes |
| 4314 | 074 | 6441 | Yes | Yes |
| 4317 | 050 | 8589 | No | Yes |
| 4317 | 070 | 8589 | Yes | Yes |
| 4317 | 072 | 7515 | Yes | Yes |
| 4317 | 074 | 6441 | Yes | Yes |
| 4318 | 050 | 17548 | No | Yes |
| 4318 | 070 | 17548 | Yes | Yes |
| 4318 | 072 | 15354 | Yes | Yes |
| 4318 | 074 | 13161 | Yes | Yes |
| 4324 | 050 | 17548 | No | Yes |
| 4324 | 070 | 17548 | Yes | Yes |
| 4324 | 072 | 15354 | Yes | Yes |
| 4324 | 074 | 13161 | Yes | Yes |

**Note**: 4314 and 4324 are Ultra-SCSI DASD; 4317 and 4318 are SCSI Wide-Ultra2 DASD.

A minimum of four drives of the same capacity are needed to protect them with RAID-5 protection. A maximum of four arrays are allowed per controller, with a maximum of ten drives allowed per array. All drives in an array must be of the same capacity. Parity is spread on four or eight drives.

### 9728 Storage Device Controller

The #9728 is a Storage controller for up to 5 internal disk units installed in the PCI System Unit. The 9728 does NOT have a Write Cache and does NOT support RAID. The 9728 also supports mixing different internal disk features on the same controller**.**

### 9728 Supported DASD Models

*Table C.8. 9728 Supported DASD Models*

| DASD | Model | MB/arm | RAID | Write Cache |
|---|---|---|---|---|
| 6606 | 030 | 1967 | No | No |
| 6806 | 030 | 1967 | No | No |
| 6607 | 030 | 4194 | No | No |
| 6807 | 030 | 4194 | No | No |
| 6713 | 030 | 8589 | No | No |
| 6813 | 030 | 8589 | No | No |
| 6714 | 030 | 17548 | No | No |

**Note**: 6806, 6807, 6813, and 6714 are Ultra-SCSI DASD

### 9767 Base PCI Disk Controller

The #9767 is a Storage controller for up to 6 internal disk units installed in the PCI System Unit. The 9767 does NOT have a Write Cache and does NOT support RAID. The 9767 also supports mixing different internal disk features on the same controller.

## 9767 Supported DASD Models

| *Table C.9. 9767 Supported DASD Models* | | | | |
|---|---|---|---|---|
| **DASD** | **Model** | **MB/arm** | **RAID** | **Write Cache** |
| 4314 | 030 | 8589 | No | No |
| 4317 | 030 | 8589 | No | No |
| 4318 | 030 | 17548 | No | No |
| 4324 | 030 | 17548 | No | No |
| **Note**: 4314 and 4324 are Ultra-SCSI DASD; 4317 and 4318 are SCSI Wide-Ultra2 DASD | | | | |

# Appendix D. CPW, CIW and MCU Values for iSeries

This chapter details the system capacities based upon the following performance workloads:

- Commercial Processing Workload (CPW). For a detailed description, refer to *Appendix A, "CPW Benchmark Description"*. CPW values are relative system performance metrics and reflect the relative system capacity for the CPW workload. CPW 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 properly reflect specific workloads other than CPW because of differing detailed characteristics (e.g., cache miss ratios, average cycles per instruction, software contention, type and number of disk devices, number of work station controllers, amount of memory, and the application being run). The CPW values shown in the tables are based on IBM internal tests. Actual performance in a customer environment will vary.
- Mail and Calendar Users (MCU). For a detailed description, refer to *Chapter 11, "Domino for iSeries"*. The MCU values represent the number Domino users executing the Mail and Calendaring Workload that are supported when the system CPU is at 70%. These values provide a better means to compare Domino capacity for various iSeries servers than does the CPW rating because MCU is computed using a Domino-specific workload. The Mail and Calendaring Workload was measured with an average of 3000 users per Domino partition.
- Compute Intensive Workload (CIW). The CIW values are application compute intensive projections based upon the characteristics of internal workloads like Domino mail and calendar and SAP. CIW values can be used in capacity planning analysis with the same cautions that were given above. The IBM Workload Estimator for iSeries 400 should be your first choice for sizing systems. See *Appendix B,"AS/400 Sizing"* for more information.

  The CIW is meant to depict a workload that has the following characteristics:
  - The majority of the system processing time is spent in the user application instead of system services. For example, MCU spends about 80% of the total processing time in the application code.
  - Compute intensive applications tend to be considerably less I/O intensive than most commercial application processing such as depicted by CPW. Therefore, cache miss rates are low and there is little or no I/O contention.

For additional CPW values, see the *IBM AS/400 Advanced 36 Performance Capabilities Reference*.

## D.1 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 12/00)

See the chapter, **iSeries RISC Server Model Performance Behavior,** for a description of the performance highlights of the new Dedicated servers for Domino models.

## D.1.1 Model 8xx Servers

| *Table D.1.1.1 Model 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 820-0150 (none) | 600 | 2 MB | 1 | 1100 | 0 | 385 | 3110 |
| 820-0151 (none) | 600 | 4 MB | 2 | 2350 | 0 | 840 | 6660 |
| 820-0152 (none) | 600 | 4 MB | 4 | 3700 | 0 | 1670 | 11810 |
| 820-2435 (1521) | 600 | 2 MB | 1 | 600 | 35 | 200 | 1620 |
| 820-2435 (1522) | 600 | 2 MB | 1 | 600 | 70 | 200 | 1620 |
| 820-2435 (1523) | 600 | 2 MB | 1 | 600 | 120 | 200 | 1620 |
| 820-2435 (1524) | 600 | 2 MB | 1 | 600 | 240 | 200 | 1620 |
| 820-2436 (1521) | 600 | 2 MB | 1 | 1100 | 35 | 385 | 3110 |
| 820-2436 (1522) | 600 | 2 MB | 1 | 1100 | 70 | 385 | 3110 |
| 820-2436 (1523) | 600 | 2 MB | 1 | 1100 | 120 | 385 | 3110 |
| 820-2436 (1524) | 600 | 2 MB | 1 | 1100 | 240 | 385 | 3110 |
| 820-2436 (1525) | 600 | 2 MB | 1 | 1100 | 560 | 385 | 3110 |
| 820-2437 (1521) | 600 | 4 MB | 2 | 2350 | 35 | 840 | 6660 |
| 820-2437 (1522) | 600 | 4 MB | 2 | 2350 | 70 | 840 | 6660 |
| 820-2437 (1523) | 600 | 4 MB | 2 | 2350 | 120 | 840 | 6660 |
| 820-2437 (1524) | 600 | 4 MB | 2 | 2350 | 240 | 840 | 6660 |
| 820-2437 (1525) | 600 | 4 MB | 2 | 2350 | 560 | 840 | 6660 |
| 820-2437 (1526) | 600 | 4 MB | 2 | 2350 | 1050 | 840 | 6660 |
| 820-2438 (1521) | 600 | 4 MB | 4 | 3700 | 35 | 1670 | 11810 |
| 820-2438 (1522) | 600 | 4 MB | 4 | 3700 | 70 | 1670 | 11810 |
| 820-2438 (1523) | 600 | 4 MB | 4 | 3700 | 120 | 1670 | 11810 |
| 820-2438 (1524) | 600 | 4 MB | 4 | 3700 | 240 | 1670 | 11810 |
| 820-2438 (1525) | 600 | 4 MB | 4 | 3700 | 560 | 1670 | 11810 |
| 820-2438 (1526) | 600 | 4 MB | 4 | 3700 | 1050 | 1670 | 11810 |
| 820-2438 (1527) | 600 | 4 MB | 4 | 3700 | 2000 | 1670 | 11810 |
| 830-2400 (1531) | 400 | 2 MB | 2 | 1850 | 70 | 580 | 4490 |
| 830-2400 (1532) | 400 | 2 MB | 2 | 1850 | 120 | 580 | 4490 |
| 830-2400 (1533) | 400 | 2 MB | 2 | 1850 | 240 | 580 | 4490 |
| 830-2400 (1534) | 400 | 2 MB | 2 | 1850 | 560 | 580 | 4490 |
| 830-2400 (1535) | 400 | 2 MB | 2 | 1850 | 1050 | 580 | 4490 |
| 830-2402 (1531) | 540 | 4 MB | 4 | 4200 | 70 | 1630 | 10680 |
| 830-2402 (1532) | 540 | 4 MB | 4 | 4200 | 120 | 1630 | 10680 |
| 830-2402 (1533) | 540 | 4 MB | 4 | 4200 | 240 | 1630 | 10680 |
| 830-2402 (1534) | 540 | 4 MB | 4 | 4200 | 560 | 1630 | 10680 |
| 830-2402 (1535) | 540 | 4 MB | 4 | 4200 | 1050 | 1630 | 10680 |
| 830-2402 (1536) | 540 | 4 MB | 4 | 4200 | 2000 | 1630 | 10680 |
| 830-2403 (1531) | 540 | 4 MB | 8 | 7350 | 70 | 3220 | 20910 |
| 830-2403 (1532) | 540 | 4 MB | 8 | 7350 | 120 | 3220 | 20910 |
| 830-2403 (1533) | 540 | 4 MB | 8 | 7350 | 240 | 3220 | 20910 |
| 830-2403 (1534) | 540 | 4 MB | 8 | 7350 | 560 | 3220 | 20910 |
| 830-2403 (1535) | 540 | 4 MB | 8 | 7350 | 1050 | 3220 | 20910 |
| 830-2403 (1536) | 540 | 4 MB | 8 | 7350 | 2000 | 3220 | 20910 |
| 830-2403 (1537) | 540 | 4 MB | 8 | 7350 | 4550 | 3220 | 20910 |
| 840-2461 (1540) | 600 | 16 MB | 24 | 20200 | 120 | 10950 | 77800 |
| 840-2461 (1541) | 600 | 16 MB | 24 | 20200 | 240 | 10950 | 77800 |
| 840-2461 (1542) | 600 | 16 MB | 24 | 20200 | 560 | 10950 | 77800 |

| *Table D.1.1.1 Model 8xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2461 (1543) | 600 | 16 MB | 24 | 20200 | 1050 | 10950 | 77800 |
| 840-2461 (1544) | 600 | 16 MB | 24 | 20200 | 2000 | 10950 | 77800 |
| 840-2461 (1545) | 600 | 16 MB | 24 | 20200 | 4550 | 10950 | 77800 |
| 840-2461 (1546) | 600 | 16 MB | 24 | 20200 | 10000 | 10950 | 77800 |
| 840-2461 (1547) | 600 | 16 MB | 24 | 20200 | 16500 | 10950 | 77800 |
| 840-2461 (1548) | 600 | 16 MB | 24 | 20200 | 20200 | 10950 | 77800 |

Note: 830 models were first available in V4R5.

## D.1.2 Model 2xx Servers

| *Table D.1.2.1 Model 2xx Servers* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 270-2431 (1518) | 540 | n/a | 1 | 465 | 30 | 185 | 1490 |
| 270-2432 (1516) | 540 | 2 MB | 1 | 1070 | 0 | 380 | 3070 |
| 270-2432 (1519) | 540 | 2 MB | 1 | 1070 | 50 | 380 | 3070 |
| 270-2434 (1516) | 600 | 4 MB | 2 | 2350 | 0 | 840 | 6660 |
| 270-2434 (1520) | 600 | 4 MB | 2 | 2350 | 70 | 840 | 6660 |

## D.1.3 V5R1 Dedicated Server for Domino

| *Table D.1.3 .1 Dedicated Servers for Domino* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPUs | NonDomino CPW | Interactive CPW | Processor CIW | MCU |
| 270-2452 (none) | 540 | 2 MB | 1 | 100 | 0 | 380 | 3070 |
| 270-2454 (none) | 600 | 4 MB | 2 | 240 | 0 | 840 | 6660 |
| 820-2456 (none) | 600 | 2 MB | 1 | 120 | 0 | 385 | 3110 |
| 820-2457 (none) | 600 | 4 MB | 2 | 240 | 0 | 840 | 6660 |
| 820-2458 (none) | 600 | 4 MB | 4 | 380 | 0 | 1670 | 11810 |

### D.1.4 Capacity Upgrade on Demand Models

New in V4R5 (12/00) , 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 BASE 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.

| | BASE 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 (12/00)

#### D.1.4.1 CPW Values and Interactive Features for CUoD Models

The following tables list only the processor CPW value for the BASE number of processors as well as a processor CPW value that represents the full capacity of the server for all processors active (BASE + ON DEMAND). If you require CPW values associated with each incremental processor activation, you could calculate the approximate CPW value associated with each processor by subtracting the maximum Processor CPW for the server from the Processor CPW published for the BASE number of processors and then dividing by the actual number of ON DEMAND 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**

| *Table D.1.4.1.1 V5R1 Capacity Upgrade on Demand Models* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2352 (1540) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 120 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1541) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 240 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1542) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 560 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1543) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 1050 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1544) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 2000 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1545) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 4550 | 3850 - 5700 | 27400 - 40500 |
| 840-2352 (1546) | 600 | 16 MB | 8 - 12 | 9000 - 12000 | 10000 | 3850 - 5700 | 27400 - 40500 |
| 840-2353 (1540) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 120 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1541) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 240 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1542) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 560 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1543) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 1050 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1544) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 2000 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1545) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 4550 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1546) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 10000 | 5700 - 8380 | 40500 - 59600 |
| 840-2353 (1547) | 600 | 16 MB | 12 - 18 | 12000 - 16500 | 16500 | 5700 - 8380 | 40500 - 59600 |
| 840-2354 (1540) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 120 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1541) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 240 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1542) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 560 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1543) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 1050 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1544) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 2000 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1545) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 4550 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1546) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 10000 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1547) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 16500 | 8380 - 10950 | 59600 - 77800 |
| 840-2354 (1548) | 600 | 16 MB | 18 - 24 | 16500 - 20200 | 20200 | 8380 - 10950 | 59600 - 77800 |

| *Table D.1.4.1.2 V4R5 Capacity Upgrade on Demand Models (12/00)* | | | | | | | |
|---|---|---|---|---|---|---|---|
| Model | Chip Speed MHz | L2 cache per CPU | CPU Range | Processor CPW | Interactive CPW | Processor CIW | MCU |
| 840-2416 (1540) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 120 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1541) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 240 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1542) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 560 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1543) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 1050 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1544) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 2000 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1545) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 4550 | 3100 - 4590 | 22000 - 32600 |
| 840-2416 (1546) | 500 | 8 MB | 8 - 12 | 7800 - 10000 | 10000 | 3100 - 4590 | 22000 - 32600 |
| 840-2417 (1540) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 120 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1541) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 240 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1542) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 560 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1543) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 1050 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1544) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 2000 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1545) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 4550 | 4590 - 6750 | 32600 - 48000 |
| 840-2417 (1546) | 500 | 8 MB | 12 - 18 | 10000 - 13200 | 10000 | 4590 - 6750 | 32600 - 48000 |
| 840-2419 (1540) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 120 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1541) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 240 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1542) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 560 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1543) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 1050 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1544) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 2000 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1545) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 4550 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1546) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 10000 | 6750 - 8820 | 48000 - 62700 |
| 840-2419 (1547) | 500 | 8 MB | 18 - 24 | 13200 - 16500 | 16500 | 6750 - 8820 | 48000 - 62700 |

## D.2 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. See the chapter, **AS/400 RISC Server Model Performance Behavior,** for a description of the performance highlights of these new models.

### D.2.1 AS/400e Model 8xx Servers

***Table D.2.1 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** |

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW** |
|---|---|---|---|---|---|
| 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** |
| | | | | | |
| 840-2418 (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** |

## D.2.2 Model 2xx Servers

| *Table D.2.2.1 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** |

## D.2.3 Dedicated Server for Domino

| *Table D.2.3.1 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** |

## D.2.4 SB Models

| *Table D.2.4.1 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.

## D.3 V4R4 Additions

The Model 7xx is new in V4R4. Also in V4R4 are the Model 170s features 2289 and 2388 were added. See the chapter, **AS/400 RISC Server Model Performance Behavior,** for a description of the performance highlights of these new models.

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.

### D.3.1 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 D.3.1.1 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 |
| 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 |

| Model | Chip Speed MHz | L2 cache per CPU | CPUs | **Processor CPW** | **Interactive CPW (Knee)** | Interactive CPW (Max) |
|---|---|---|---|---|---|---|
| 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 |

## D.3.2 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 D.3.2.1 Current Model 170 Servers*** ***(Note: 2289 amd 2388 are two new Northstar models)***

| 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 D.3.2.2 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 D.3.2 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 |

## D.4 AS/400e Model Sxx Servers

For AS/400e servers the knee of the curve is about 1/3 the maximum interactive CPW value.

***Table D.4.1 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 |

## D.5 AS/400e Custom Servers

For custom servers the knee of the curve is about 6/7 maximum interactive CPW value.

***Table D.5.1 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 |

## D.6 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 D.6.1 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 D.6.2 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 |

## D.7 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 D.7.1 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 |

## D.8 AS/400 Models 4xx, 5xx and 6xx Systems

| *Table D.8.1 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 D.8.2 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 |

## D.9 AS/400 CISC Model Capacities

| *Table D.9.1 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 D.9.2 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 D.9.3 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 D.9.4 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 D.9.5 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 D.9.6 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 D.9.7 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 D.9.8 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 |