IBM @server

iSeries Performance Capabilities Reference
Version 5, Release 2

January 2003

This document is intended for use by qualified performance related programmers or analysts from IBM, IBM Business Partners and IBM customers using iSeries servers. 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 2. 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."

Eighteen Edition (January 2003) SC41-0607-06

This edition applies to Version 5, Release 2 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 5 Release 1 and Version 4 Release 5 Performance Capabilities Guides 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

Table of Contents

V5R2 Performance Capabilities Reference - January 2003

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/400	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.

iSeries Performance Capabilities   **10**

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 **January 2003 edition** of the V5R2 Performance Capabilities Reference Guide is an update to the June 2002 edition to reflect new product functions announce on January 28, 2003, with available through the remainder of 2003. This January 2003 edition supersedes the V5R2 June 2002 edition.

This January 2003 Edition include new performance information on new iSeries Model 800, 810, 825, 870 and 890 CUoD models, disk and I/O, WebSphere 5.0, WebSphere Express, WebSphere Portal Server, Web-Facing, security, Web Serving, WLE, PM eServer iSeries, TCP/IP performance and many other topics.

The wide variety of applications available makes it extremely difficult to describe a "typical" workload. The data in this document is the result of measuring or modeling certain application programs in very specific and unique configurations, and should not be used to predict specific performance for other applications. The performance of other applications can be predicted using a system sizing tool such as IBM eServer Workload Estimator or Patrol for iSeries - Predict (refer to Chapter 22 for more details on Workload Estimator).

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

V5R2 continues to enhance the iSeries value proposition - the best melding of a superior operating system with 64-bit RISC hardware. The performance of V5R2 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 5250 OLTP, web-facing, client serving, batch, Domino, Java, Linux, business intelligence, and web (WebSphere) serving applications.

The iSeries Model 890 32-way processor was made available in V5R2, increasing performance of iSeries by 1.85X over i840 24-way available in V5R1. V5R2 now supports new Model 800, 810, 825, 870 and 890 Capacity Upgrade on Demand servers. In addition, the new models offer attractive editions (Standard and Enterprise) with enhanced software for multiple operating systems, web modernization, dynamic partioning and 5250 OLTP processing flexibility. The new models and software offering editions offer significant price/performance improvements.

The primary V5R2 functional performance items are:

- New models for entry, middle and high-end growth, with mainframe class scaling up to 32-way processors with up to 37,400 CPW (Commercial Processing Workload), 16,700 CIW (Compute Intensive Workload) metrics and up to 108,900 Mail/Calendar Users in Domino
- Improved V5R2 CPW values, extremely fast processors (up to 1.3 GHz), large L2/L3 caches (up to 534MB), more memory (up to 256GB)
- New Capacity Upgrade on Demand server models from 3 to 6-way, 8 to 16-way, 16 to 24-way and 24 to 32-way models for quick nondisruptive processor activation when needed.   These new models offer both permanent and temporary processor capacity on demand options.
- New PCI-X , I/O processors and high capacity 15k rpm disk units enhance function and capacity
- Improved PC performance on PCI Integrated xSeries™ Server for iSeries
- Model 890 supports a new bus with an aggregate switch speed of up to 205GB per second (up to 4.5X performance improvement over i840 bus)
- 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 supports latest standards and enhanced query support with DB2 UDB
- Enhanced support for continuous availability clustering and disk swapping
- New secure Enterprise-class TCP/IP support and dramatic Gigabit Ethernet performance
- Enhanced server-side Java performance measured by VolanoMark and secure web serving performance by SPECWeb99SSL benchmarks
- Delivers enhanced self-managing solutions with industry's first Enterprise Identity Mapping (EIM) for single signon and support for storage virtualization with switched disk clustering
- WebSphere and secure web server performance improvements with adaptive e-transaction management services and price/performance improvements with WebSphere-Express offering
- Linux supported in secondary partitions of LPAR

OS/400 V5R2 operating system with February 2003 PTF's enables the new iSeries 800, 810, 825, 870 and new 890 models. Customers who wish to remain with their existing hardware but want to move to the V5R2 operating system may find functional and performance improvements. Version 5 Release 2 OS/400 continues to help protect the customer's investment while providing more function, growth, capacity, performance and better price/performance over previous versions. The primary public performance information 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 and AS/400 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	

Note that printer work that passes through twinax media is treated as interactive, even though there is no "user interface". This is true regardless of whether the printer is working in dedicated mode or is printing spool files from an out queue. Printer activity that is routed over a LAN through a PC print controller are not considered to be interactive.

This explanation is different than that found in previous versions of this document. Previous versions indicated that spooled work would not be considered to be interactive and were in error.

As of January 2003, 5250 On-line Transaction Processing (OLTP) replaces the term "interactive" when referencing interactive CPW or interactive capacity. Also new in 2003, when ordering a iSeries server, the customer must choose between a Standard Package and an Enterprise Package in most cases. The Standard Packages comes with zero 5250 CPW and 5250 OLTP workloads are not supported. However, the Standard Package does support a limited 5250 CPW for a system administrator to manage various aspects of the server. Multiple administrative jobs will quickly exceed this capability. The Enterprise Package does not have any limits relative to 5250 OLTP workloads. In other words, 100% of the server capacity is available for 5250 OLTP applications whenever you need it.

5250 OLTP applications can be run after running the WebFacing Tool of IBM Websphere Development Studio for iSeries and will require no 5250 CPW if on V5R2 and using model 800, 810, 825, 870, or 890 hardware.

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, two new metrics were added to the data generated by Collection Services to report the system's interactive CPU utilization (ref file QAPMSYSCPU). The first metric (SCIFUS) is the interactive utilization - an average for the interval. Since average utilization does not indicate potential problems associated with peak activity, a second metric (SCIFTE) reports the amount of interactive utilization that occurred above threshold. 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.

Also in V4R5, a new operator message, CPI1479, was introduced for when the system has consistently exceeded the purchased interactive capacity on the system. The message is not issued every time the capacity is reached, but it will be issued on an hourly basis if the system is consistently at or above the limit. In V5R2, this message may appear slightly more frequently for 8xx systems, even if there is no change in the workload. This is because the message event was changed from a point that was beyond the purchased capacity to the actual capacity for these systems in V5R2.

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 to understand what tasks have contributed to the system's interactive utilization and the CPU utilized by all interactive tasks. Note: the system's interactive capacity utilization may not be equal to the utilization of all interactive tasks. Reasons for this are discussed in Section 2.10, *Managing Interactive Capacity*.

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 iSeries 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
- Dedicated Server for Domino (DSD) Performance Behavior

2.1.3 V5R2 and V5R1

There were several new iSeries 8xx and 270 server model additions in V5R1 and the i890 in V5R2. However, with the exception of the DSD models, the underlying server behavior did not change from

V4R5. All 27x and 8xx models, including the new i890 utilize the same server behavior algorithm that was announced with the first 8xx models supported by V4R5. For more details on these new models, please refer to *Appendix C, . CPW, CIW and MCU Values for iSeries.*

Five new iSeries 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 In V4R5 - V5R2

Beginning with V4R5, all 2xx, 8xx and SBx model servers 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 cases where interactive demand exceeds the limitations of the interactive capacity for an extended time (for example: 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 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 iSeries 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 (see section 2.2.3 Existing Model section for 7xx algorithm).

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 tend 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 eServer Workload Estimator to help you select the correct system (see URL: http://www.ibm.com/iseries/support/estimator) .

2.2.3 Existing Older Models

Server model behavior applies to:
- AS/400 Advanced Servers
- AS/400e servers
- AS/400e custom servers
- AS/400e model 150
- iSeries model 170
- iSeries model 7xx

Relative performance measurements are derived from commercial processing workload (CPW) on iSeries and 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.
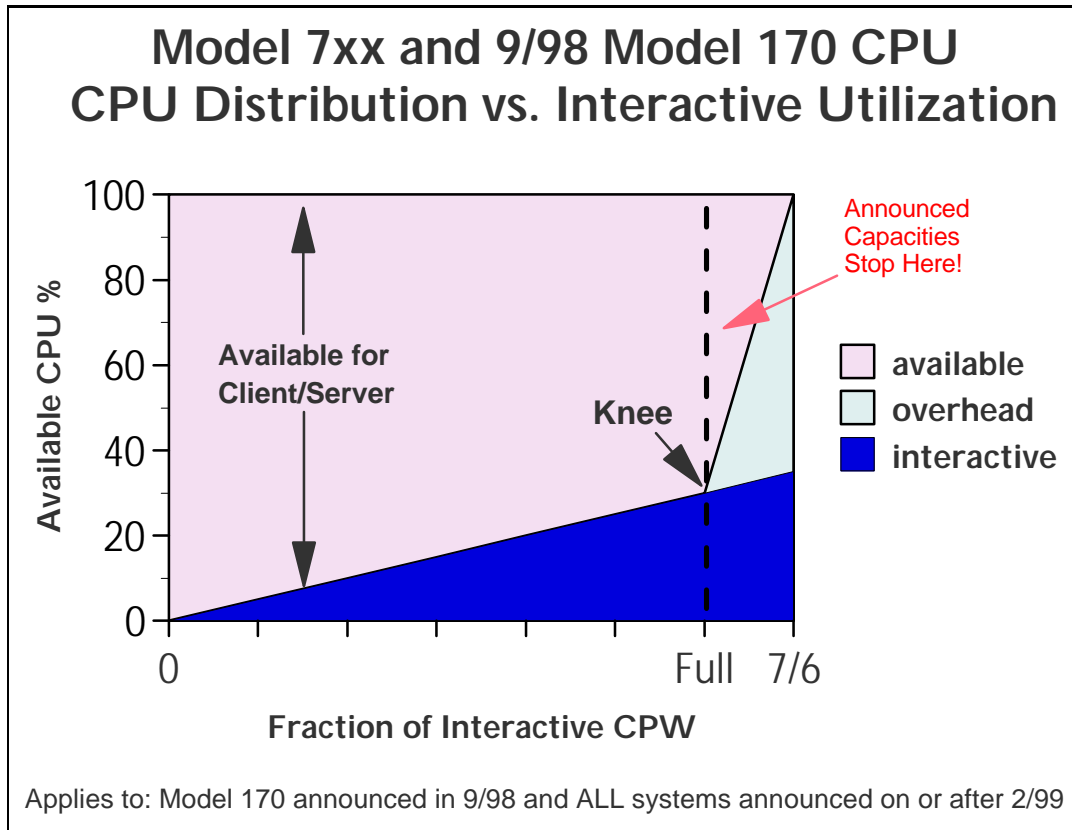
Figure 2.1. Server Model behavior

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 iSeries 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:
    - iSeries 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
    - iSeries 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.
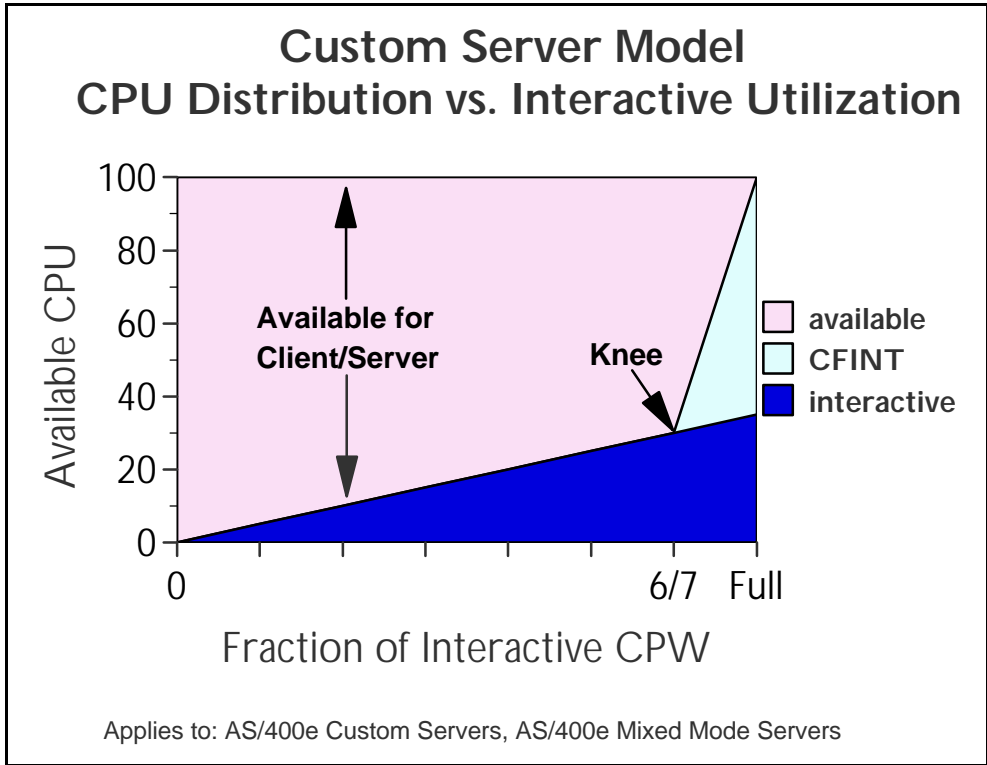
Figure 2.2. Custom Server Model behavior
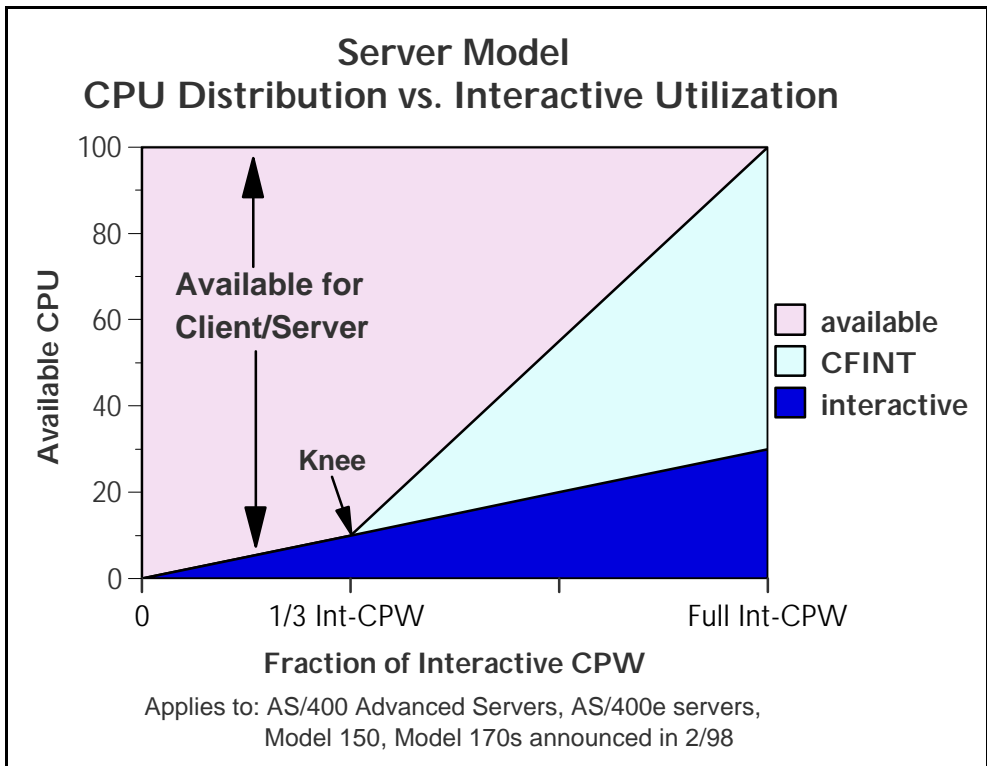
Figure 2.3. Server Model behavior

2.4 Performance Highlights of 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:
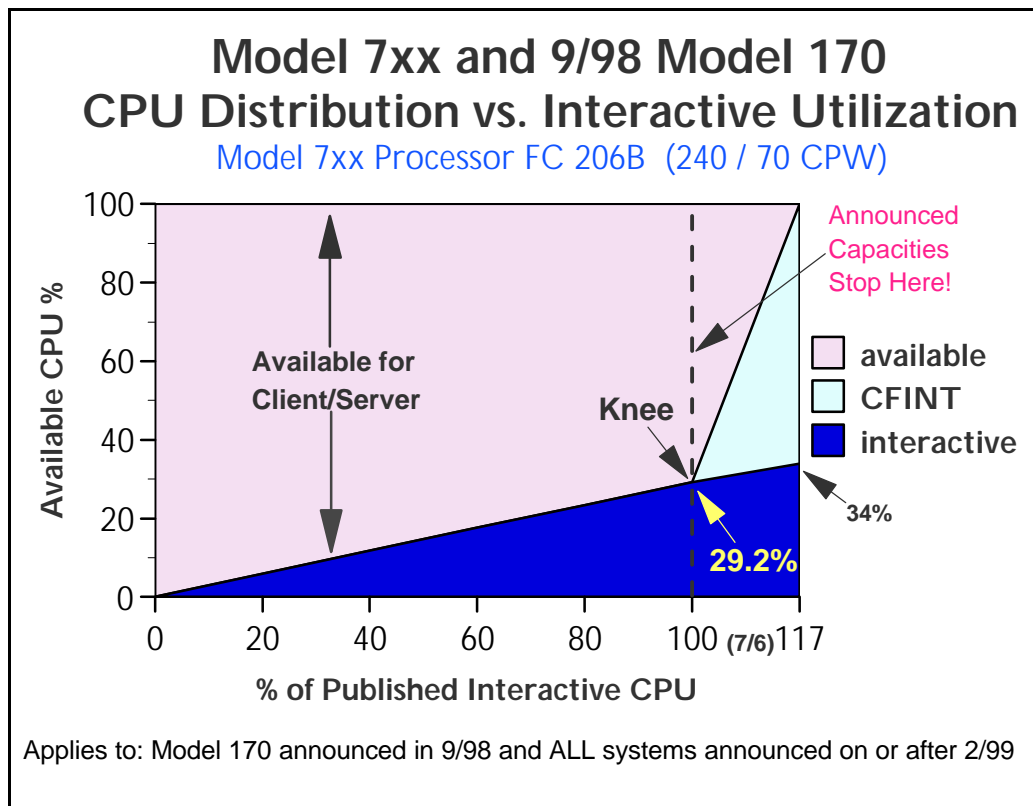
Figure 2.4. Model 7xx Utilization Example

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 Model 170 Servers

iSeries Dedicated Server for Domino models will be generally available on September 24, 1999. Please refer to Section 2.13, *iSeries 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.

Previous vs. Current AS/400e server 170 Performance

CPW Values chart with categories:

Previous *
- 2159: Interactive 16, Processor 73
- 2160: Interactive 23, Processor 114
- 2164: Interactive 29, Processor 210
- 2176: Interactive 40, Processor 319
- 2183: Interactive 67, Processor 319

Current
- 2289: Interactive 15, Processor 50
- 2290: Interactive 20, Processor 73
- 2291: Interactive 25, Processor 115
- 2292: Interactive 30, Processor 220
- 2385: Interactive 50, Processor 460
- 2386: Interactive 70, Processor 460
- 2388: Interactive 70, Processor 1090

Legend: Interactive, Processor

* Unconstrained V4R2 rates

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 Collection Services 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.

Affects of Server Dynamic Tuning

Server Dynamic Tuning - .
High "Server" Demand

- available
- interactive

Available CPU

Available for Client/Server

Knee

0   1/3 Int-CPW   Full Int-CPW
Fraction of Interactive CPW

- With sufficient batch or client/server load, Interactive is constrained to the "knee-level" by priority degradation

- Interactive suffers poor response times

Server Dynamic Tuning
Mixed "Server" Demand

- available
- O.H. or Server
- Int. or Server
- interactive

Available CPU

Available for Client/Server

Knee

0   1/3 Int-CPW   Full Int-CPW
Fraction of Interactive CPW

- Without high "server" demand, Interactive allowed to grow to limit

- Overhead introduced just as when Dynamic Priority Adjust is turned off

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 Collection Services used in conjunction with the Performance Tools program product. The monitor collects system data as well as 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 various performance data base files.
   Note: data is written to these files based on sample interval (Smallest is 5 minutes, default is 15 minutes). This data is an average for the duration of a measurement interval.

1. The first metric of interest is how much of the system's interactive capacity has been used. The file QAPMSYSCPU field SCIFUS contains the amount of interactive feature CPU time used. This metric became available with Collection Services in V4R5.

2. Even though average CPU may be reasonable your interactive workload may still be exceeding limits at times. The file QAPMSYSCPU field SCIFTE contains the amount of time the interactive threshold was exceeded during the interval. This metric became available with Collection Services in V4R5.

3. To determine what jobs are responsible for interactive feature consumption, you can look at the data in QAPMJOBL (Collection Services) or QAPMJOBS (Performance Monitor):
   - If using Collection Services on a V5R1 or later system, those jobs which the machine considers to be interactive are indicated by the field JBSVIF ='1'. These are all jobs that could contribute to your interactive feature utilization.

- In all cases you can examine the jobs that are considered interactive by OS/400 as indicated by field JBTYPE = "I". Although not totally accurate, in most cases this will provide an adequate list of jobs that contributed to interactive utilization.

There are other means for determining interactive utilization. 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.

Finally, the functions of PM400 can also show the same type of data that Collection Services 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 iSeries 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, Collection Services 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 may show the current activity for the service jobs and/or 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 than 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.

- Lastly when only a single interactive job is running, the machine grants an exemption and does not include this job's activity in the interactive feature utilization.

There are two key ideas in the statements above. First, if the workload has a significant component that is related to queries or there is a single interactive job running, it will be possible to show an interactive job utilization in the performance tools that is significantly higher than what would be assumed and reported 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

 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 Collection Services during selected time periods and review the first page of the Performance Tools for iSeries 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" in Collection Services or the 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 for iSeries 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 Upgrade Considerations for Interactive Capacity

When upgrading a system to obtain more processor capacity, it is important to consider upgrading the interactive capacity, even if additional interactive work is not planned. Consider the following hypothetical example:
- The original system has a processor capacity of 1000 CPW and an interactive capacity of 250 ICPW
- The proposed upgrade system has a processor capacity of 4000 CPW and also offers an interactive capacity of 250 ICPW.
- On the original system, the interactive capacity allowed 25% of the total system to be used for interactive work. On the new system, the same interactive capacity only allows 6.25% of the total system to be used for interactive work.
- Even though the total interactive capacity of the system has not changed, the faster processors (and likely larger memory and faster disks) will allow interactive requests to complete more rapidly, which can cause greater spikes of interactive demand.
- So, just as it is important to consider balancing memory and disk upgrades with processor upgrades, optimal performance may also require an interactive capacity upgrade when moving to a new system.

2.13 iSeries for Domino and Dedicated Server for Domino Performance Behavior

Included in the new V5R2 February 2003 iSeries models are five *iSeries for Domino* offerings. These include three i810 and two i825 models. The iSeries for Domino offerings are specially priced and configured for Domino workloads. There are no processing guidelines for the iSeries for Domino offerings as with non-Domino processing on the Dedicated Server for Domino models. With the iSeries for Domino offerings the full amount of DB2 processing is available, and it is no longer necessary to have Domino processing active for non-Domino applications to run well. Please refer to Chapter 11 for additional information on Domino performance in iSeries, and Appendix C for information on performance specifications for iSeries servers.

For existing iSeries servers, OS/400 V5R2 (both the June 2002 and the updated February 2003 version) will exhibit similar performance behavior as V5R1 on the Dedicated Server for Domino models. The following discussion of the V5R1 Domino-complimentary behavior is applicable to V5R2.

Five new DSD models were announced with V5R1. These included 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 was 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 was available after September 28, 2001 with a 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 iSeries 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 C for performance specifications for 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 was 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 iSeries, 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 iSeries, 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 LPAR web at:
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 utilization 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 Linux for iSeries web site at:
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 iSeries. For example, PATROL for iSeries - Predict from BMC Software, Inc. * (PID# 5620FIF) 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.

* BMC Software, the BMC Software logos and all other BMC Software products including PATROL for iSeries - Predict are registered trademarks or trademarks of BMC Software, Inc.

Chapter 4. DB2 UDB for iSeries Performance

There are many factors which affect overall DB2 UDB performance and much detailed information available on the topics. This chapter provides a summary of the new features of DB2 UDB for V5R2 and some key topics on the performance of DB2 UDB. General information and some recommendations for improving performance are included along with links to the latest information on these topics. Also included is a section of performance references for DB2 UDB.

4.1 New for V5R2 SQL Query Engine

In V5R2 major enhancements, entitled SQL Query Engine (SQE), were implemented in DB2 UDB for iSeries. SQE encompasses changes made in the following areas:
- SQL query optimizer
- SQL query engine
- Database statistics

A subset of the read-only SQL queries will be able to take advantage of these enhancements in V5R2.

SQE Optimizer

The SQL query optimizer has been enhanced with new optimization capabilities implemented in object oriented technology. This object oriented framework implements new optimization techniques and allows for future extendibility of the optimizer. Among the new capabilities of the optimizer are enhanced query access plan costing. For queries which can take advantage of the SQE enhancements, more information may be used in the query plan costing phase than was available to the optimizer previously. The optimizer may now use newly implemented database statistics to make more accurate decisions when choosing the query access plan. Also, the enhanced optimizer may more often select plans using hash tables and sorted partial result lists to hold partial query results during query processing, rather than selecting access plans which build temporary indexes. With less reliance on temporary indexes the SQE optimizer is able to select more efficient plans which save the overhead of building temporary indexes and more fully take advantage of single-level store. The optimizer changes were designed to create efficient query access plans for the enhanced database engine.

SQE Query Engine

The database engine is the part of the database implementation which executes the access plan produced by the query optimizer. It accesses the data, processes it, and returns the SQL query results. The new engine enhancements, the SQE database engine, employ state of the art object oriented implementation. The SQE database engine was developed in tandem with the SQE optimization enhancements to allow for an efficient design which is readily extendable. Efficient new algorithms for the data access methods are used in query processing by the SQE engine.

The basic data access algorithms in SQE are designed to take full advantage of the iSeries single-level store to give the fastest query response time. The algorithms reduce I/O wait time by making use of available main memory and aggressively reading data from disk into memory. The goal of the data read-ahead algorithms is that the data is in memory when it is needed. This is done through the use of asynchronous I/Os. SQL queries which access large amounts of data may see a considerable improvement in the query run time. This may also result in higher peak disk utilization.

To show performance changes in the table scan access method the following query was run over a range of file sizes: select fieldb from tablea where fielda < value. Value was chosen such that 20% of the records in the table were selected. There is an index over fielda, however the optimizer determined that the most efficient access method to use was scanning the whole table. The results with the SQE enhancements versus without the SQE enhancements (CQE) are shown below in Table 4.1.

Table 4.1 Table Scan Access Method Performance

File Size	SQE Run Time (sec)	CQE Run Time (sec)	SQE CPU Time (sec)	CQE CPU Time (sec)	NQE Avg % Disk Util	CQE Avg % Disk Util
2 MB	0.5	0.6	0.1	0.1	0.8	0.0
20 MB	0.9	1.3	0.2	0.3	7.6	1.2
50 MB	1.5	2.8	0.4	0.6	19.3	2.5
100 MB	2.2	4.6	0.7	1.1	27.5	3.0
250 MB	4.8	10.8	1.6	3.1	41.1	4.4
500 MB	8.9	21.8	3.1	6.5	48.6	4.3
1 GB	17.6	42.7	6.2	11.6	52.6	4.5
2 GB	45.4	84.4	12.5	24.4	44.4	3.9
5 GB	84.3	206.3	30.6	59.3	62.9	3.9
10 GB	164.2	405.7	61.1	118.9	64.6	4.3

Notes:
- The files and indexes were purged from memory prior to running the query.
- To simulate expected V5R2 vs V5R1 performance results, the SQE vs. CQE comparison measurements were run on the same hardware (processor, memory, dasd configuration...) and using the same files and indexes. For laboratory purposes, the query was measured with the SQE enhancements using the SQE code path and without the enhancements (CQE) by invoking the CQE code path.
- The queries were measured with QQRYDEGREE=*NONE, on a dedicated iSeries LPAR with the given query running as the only job in a shared pool with paging option *CALC.
- The results may vary significantly depending on the system configuration and file layout and data characteristics.

As is shown in Table 4.1 as the file size increases the benefit of the SQE table scan access method implementation becomes more significant. Queries over small files see little improvement in run time, while those over large files see a significant improvement. This is due to the aggressive asynchronous I/Os used by SQE. This results in significantly larger dasd utilization over the query run time for the ASP on which the file resides. An additional benefit for this query with SQE is the reduction of CPU used to execute the query.

To show performance changes when using an index for selection of the records to probe from the table the following query was run over a range of file sizes: select fieldb from tablea where fielda < value. Value was chosen such that 0.25% of the records in the table were selected. There is an index over fielda. The optimizer determined that probing the index to find matching records to probe from the table was the most efficient access method to use. The results with the SQE enhancements versus without the SQE enhancements (CQE) are shown below in Table 4.2.

Table 4.2 Index Access Method Performance

File Size	SQE Run Time (sec)	CQE Run Time (sec)	SQE CPU Time (sec)	CQE CPU Time (sec)	NQE Avg % Disk Util	CQE Avg % Disk Util
2 MB	0.3	0.3	0.1	0.1	0.0	1.9
20 MB	0.6	0.6	0.1	0.1	4.7	5.6
50 MB	1.1	0.7	0.2	0.1	16.3	5.3
100 MB	1.7	1.6	0.3	0.1	22.5	29.8

250 MB	**1.9**	**2.0**	**0.3**	**0.3**	**31.8**	**25.5**
500 MB	**3.0**	**3.8**	**0.4**	**0.5**	**43.5**	**34.4**
1 GB	**5.6**	**7.2**	**0.9**	**0.9**	**55.9**	**43.1**
2 GB	**10.3**	**14.2**	**1.5**	**1.7**	**65.5**	**52.6**
5 GB	**25.1**	**37.6**	**3.5**	**4.3**	**71.3**	**52.1**
10 GB	**51.6**	**77.3**	**7.9**	**8.6**	**69.3**	**54.2**

Notes:
- The files and indexes were purged from memory prior to running the query.
- To simulate expected V5R2 vs V5R1 performance results, the SQE vs. CQE comparison measurements were run on the same hardware (processor, memory, dasd configuration...) and using the same files and indexes. For laboratory purposes, the query was measured with the SQE enhancements using the SQE code path and without the enhancements (CQE) by invoking the CQE code path.
- The queries were measured with QQRYDEGREE=*NONE, on a dedicated iSeries LPAR with the given query running as the only job in a shared pool with paging option *CALC.
- The results may vary significantly depending on the system configuration and file layout and data characteristics.

As is shown in Table 4.2 with small files the query run time is comparable for SQE and CQE. There is little to no improvement in run time. However as the file size increases run time improvements become appreciable. This is due to the asynchronous I/Os used by SQE. The average dasd utilization over the query run time for the ASP on which the file resides is larger with the SQE enhancements for large file access.

The effects of the SQE enhancements on SQL query performance will vary greatly depending on many factors. Among these factors are hardware configuration (processor, memory size, dasd configuration...), system value settings, file layout, indexes available, query options file QAQQINI settings, and the SQL queries being run. Some examples of individual query performance with the SQE enhancements vs. without the SQE enhancements (CQE) are shown below in Table 4.3. These are a set of simple queries run on one system configuration and are not meant to represent the performance expectations for any other workload.

Table 4.3 SQL Simple Query Performance Comparison

Query Text	SQE Run Time (sec)	CQE Run Time (sec)	Ratio SQE/CQE Run Time
Table Scan Queries			
select * from table1	0.16	0.27	0.60
select integer1 from table1	0.14	0.26	0.52
select integer1 from table2 where integer3 between 10000 and 20000 or integer3 between 30000 and 40000	0.09	0.20	0.46
select * from tablea	0.36	0.57	0.62
select * from tablea where field9='3/17/1991'	0.17	0.38	0.45
select * from tablea where field7*field8 <= 50000	0.32	0.45	0.72
select * from tablea where float(field1) * float(field2) <= 50000	0.21	0.43	0.49
select field1, field2, field3, field4, field5, field6 from tablej where field10 = 99999	1.26	3.68	0.34
select * from tablej where field12 = 228221 or field12 = 153572 or field12 = 240145 or field12 = 160961 order by field12,field11	1.38	3.65	0.38
select count(*) from tablej where field12 = 228221 or field12 = 153572 or field12 = 240145 or field12 = 160961	1.30	3.54	0.37
Index Access Queries			
select integer1 from table2 order by integer1	0.10	0.58	0.18
select * from table2 where integer1=23095	0.08	0.07	1.12

select field1, field2, field3, field4, field5, field6 from tablej where field1 = 99999	0.04	0.04	1.11
select * from tablej where (field3 = 223838 or field3 = 258700 or field3 = 108292 or field3 = 220440) and field2 <= 22000	0.31	0.32	0.97
select count(*) from tablej where field3 = 223838 or field3 = 258700 or field3 = 108292 or field3 = 220440	0.04	0.06	0.72
select * from tablea where field2 in(50037,50483,50560,51122,56268)	0.10	0.09	1.11
select * from tablea where field2 between 5001 and 6000	0.17	0.16	1.06
select * from tablea where field4 like 'abmaaaaaaaaaaaaaaaaaa'	0.38	0.38	0.99
select * from table2 order by integer1	0.68	1.04	0.66
select * from tablej where field3 = 223838 or field3 = 258700 or field3 = 108292 or field3 = 220440 order by field3,field2	0.63	0.67	0.95
select * from tablea order by field4	1.39	1.55	0.89
Join Queries			
select integer1, sum(integer2) from table2 group by integer1	0.07	0.38	0.17
select * from table1 join table2 on table1.integer1 = table2.integer1 where table1.integer1 between 10000 and 20000 optimize for 1 rows	1.19	2.17	0.55
select table1.integer2, sum(table2.integer4) from table1 join table2 on table1.integer1 = table2.integer5 group by table1.integer2	0.98	2.50	0.39
select * from table2 order by integer1	0.69	1.02	0.68
select * from tableb, tablec where tableb.field11 = tablec.field11 and tableb.field11>250000	3.45	3.81	0.91
select * from tableb, tablec where tableb.field2 = tablec.field2 and tableb.field2>250000	3.10	3.55	0.87
select distinct tableb.field5 from tablea, tableb, tablec, tableg where tablea.field1 = tableb.field2 and tablea.field1 = tablec.field1 and tablea.field1 = tableg.field2	9.40	123.91	0.08

Notes:
- The files and indexes were purged from memory prior to running the query.
- The measurements for the SQE results were run with the SQE enabling change as in Info APAR II13486.
- To simulate expected V5R2 vs V5R1 performance results, the SQE vs. CQE comparison measurements were run on the same hardware (processor, memory, dasd configuration...) and using the same files and indexes. For laboratory purposes, the query was measured with the SQE enhancements using the SQE code path and without the enhancements (CQE) by invoking the CQE code path.
- The queries were measured with QQRYDEGREE=*NONE, on a dedicated iSeries LPAR with the given query running as the only job in a shared pool with paging option *CALC.
- The results may vary significantly depending on the system configuration and file layout and data characteristics.

SQE Database Statistics

The third area of SQE enhancements is the collection and use of new database statistics. Efficient processing of database queries depends primarily on a query optimizer that is able to make judicious choices of access plans. The ability of an optimizer to make a good decision is critically influenced by the availability of database statistics on tables referenced in queries. In the past such statistics were automatically gathered during optimization time for columns of tables over which indexes exist. With SQE statistics on columns without indexes can now be gathered and will be used during optimization. Column statistics comprise histograms, frequent values list, and column cardinality.

With iSeries servers, the database statistics collection process is handled automatically, while on many platforms statistics collection is a manual process that is the responsibility of the database administrator. It

is rarely necessary for the statistics to be manually updated, even though it is possible to manage statistics manually. The Statistics Manager determines on what columns statistics are needed, when the statistics collection should be run and when the statistics need to be refreshed. Statistics are automatically collected as low priority work in the background, so as to minimize the impact to other work on the system. The manual collection of statistics is run with the normal user job priority.

The system automatically determines what columns to collect statistics on based on what queries have run on the system. Therefore for queries which have slower than expected performance results, a check should be made to determine if the needed statistics are available. Also in environments where long running queries are run only one time, it may be beneficial to ensure that statistics are available prior to running the queries.

Some properties of database column statistics are as follows:

- Column statistics occupy little storage, on average 8-12k per column.

- Column Statistics are gathered through one full scan of the database file for any given number of columns in the database file.

- Column statistics are maintained periodically through means of statistics refreshing mechanisms that require a full scan of the database file.

- Column statistics are packed in one concise data structure that requires few I/Os to page it into main memory during query optimization.

As stated above, statistics may have a direct effect on the quality of the access plan chosen by the query optimizer and thereby influence the end user query performance. Shown below is an illustrative example that underscores the effect of statistics on access plan selection process.

Statistic Usage Example:

Select * from T1, T2 where T1.A=T2.A and T1.B = 'VALUE1' and T2.C = 'VALUE2'

Database characteristics: indexes on T1.A and T2.A exist, NO column statistics, T1 has 100 million rows, T2 has 10 million rows. T1 is 1 GB and T2 0.1 GB

Since statistics are not available, the optimizer has to consider default estimates for selectivity of T1.B = 'VALUE1' ==> 10%   T2.C = 'VALUE2' ==> 10%

The actual estimates are T1.B = 'VALUE1' ===>10% and   T2.C = 'VALUE2' ===>0.00001%

Based on selectivity estimates the optimizer will select the following access plan

Scan(T1) - Probe (T2.A index) - > Probe (T2 Table) ---

the real cost for the above access plan plan would be approximately 8192 I/Os + 3600 I/Os **~ 11792 I/Os**

If column statistics existed on T2.C the selectivity estimate for T2.C = 'VALUE2' would be 10 rows or 0.00001%

And the query optimizer would select the following plan instead

Scan(T2) - Probe (T1.A index) - > Probe (T1 Table)

Accordingly the real cost could be calculated as follows:

819 I/Os + 10 I/Os ~ **830 I/Os. The result of having statistics on T2.C led to an access plan that is faster by order of magnitude from a case where no statistics exist.**

For more information on database statistics collection see the DB2 UDB for iSeries Database Performance and Query Optimization manual at http://publib.boulder.ibm.com/html/as400/v5r2/ic2924/index.htm?info/rzajq/rzajqmst02.htm.

SQE for V5R2 Summary

Enhancements to DB2 UDB for iSeries, called SQE, were made in V5R2. The SQE enhancements are object oriented implementations of the SQE optimizer, the SQE query engine and the SQE database statistics. In V5R2 a subset of the read-only SQL queries will be optimized and run with the SQE enhancements. The effect of SQE on performance will vary by workload and configuration. For the most recent information on SQE please see the SQE webpage on the DB2 UDB for iSeries website located at www.iseries.ibm.com/db2/sqe.html. More information on SQE for V5R2 will also be available in an upcoming redbook. Please check the redbook repository located at http://publib-b.boulder.ibm.com/Redbooks.nsf/Portals/AS400.

4.2 Indexing

Index usage can dramatically improve the performance of DB2 UDB SQL queries. For detailed information on using indexes see the white paper *Indexing Strategies for DB2 UDB for iSeries* at http://www.iseries.ibm.com/developer/bi/documents/strategy/strategy.html . The paper provides basic information about indexes in DB2 UDB for iSeries, the data structures underlying them, how the system uses them and index strategies. Also discussed are the additional indexing considerations related to maintenance, tools and methods.

Encoded Vector Indices (EVIs)

DB2 UDB on iSeries supports the Encoded Vector Index (EVI) which can be created through SQL. EVIs cannot be used to order records, but in many cases, they can improve query performance. 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 from traditional indexes.

- 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 iSeries 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.

4.3 DB2 UDB Symmetric Multiprocessing feature

Introduction

The DB2 UDB SMP feature provides application transparent support for parallel query operations on a single tightly-coupled multi-processor iSeries system (shared memory and disk). In addition, the symmetric multiprocessing (SMP) feature provides additional query optimization algorithms for retrieving data. 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. For more information on access methods which use the SMP feature and how to enable SMP see the DB2 UDB for iSeries Database Performance and Query Optimization manual at
http://publib.boulder.ibm.com/html/as400/v5r2/ic2924/index.htm?info/rzajq/rzajqmst02.htm

Decision Support Queries

The SMP feature is most useful when running decision support (DSS) queries. DSS queries which 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 iSeries 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 more than 25 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.

For more information on DSS queries and data warehousing go to the iSeries Teraplex Center home page at http://www.iseries.ibm.com/developer/bi/teraplex/ .

4.4 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.

Remote Journal Function

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.

System-Managed Access-Path Protection (SMAPP)

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 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 will minimize the performance impact while at the same time providing a reasonable and predictable recovery time and protection for key access paths. 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. 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 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 for iSeries* 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 topic *Collecting Performance Data for analysis* under *Performance* in the *iSeries Information Center*).

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.

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.

For more information on the performance impact of journaling see the redbook *Striving for Optimal Journal Performance on DB2 Universal Database for iSeries* at
http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg246286.pdf.

4.5 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.

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 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.

For more information on DB2 Multisystem refer to the DB2 Multsystem manual.

4.6 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 UDB 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.

For more information on Referential Integrity see the chapter *Insuring Data Integrity with Referential Constraints* in *DB2 Universal Database for iSeries Database Programming* manual and the redbook *Advanced Functions and Administration on DB2 Universal Database for iSeries* at http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae006bb65f/189e93edc38f03b78 52569d2005a959a?OpenDocument.

4.7 Triggers

Trigger support for DB2 UDB 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. For more information see the redbook *Stored Procedures and Triggers on DB2 Universal Database for iSeries* at
http://publib-b.boulder.ibm.com/Redbooks.nsf/9445fa5b416f6e32852569ae006bb65f/559d4ee24ce894f685256ac40061faa3?OpenDocument .

4.8 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 iSeries 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 DB2 UDB 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 Reference 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 Reference or the SQL Programming Concepts.

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 UDB 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.

4.9 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 deactivated 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.

4.10 Null Values

DB2 UDB 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 DB2 UDB for iSeries SQL Programming Concepts or the SQL Reference.

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.

4.11 Performance References for DB2 UDB

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

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

3. Information on creating efficient running queries and query performance montoring and tuning is found in the DB2 UDB for iSeries Database Performance and Query Optimization manual at http://publib.boulder.ibm.com/html/as400/v5r2/ic2924/index.htm?info/rzajq/rzajqmst02.htm. This document contains detailed information on access methods, the query optimizer, and optimizing query performance including using database monitor to monitor queries, using QAQQINI file options and using indexes.

4. 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 see the Teraplex Center's Internet home page at http://www.iseries.ibm.com/developer/bi/teraplex/. Under the White Papers and Technical Articles link is the paper *Indexing Stratgies for DB2 UDB for iSeries* at http://www.iseries.ibm.com/developer/bi/documents/strategy/strategy.html which has detailed information on using indexes to improve performance in DB2 UDB.

5. The iSeries redbooks provide performance information on a variety of topics for DB2 UDB. The redbook repository is located at http://publib-b.boulder.ibm.com/Redbooks.nsf/Portals/AS400.

Chapter 5. Communications Performance

There are many factors that affect iSeries performance in a communications environment. This chapter discusses some of the common factors and offers guidance on how to achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and 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.

This chapter focuses on communications in a non-secure environment. Many applications require network communications to be secure. Communications and cryptography, in these cases, must be considered together. See Chapter 8, "Cryptography Performance" for performance imformation about secure e-Business tranactions over a network.

Communications Performance Highlights for V5R2:

- Again in V5R2 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, V4R5 and V5R1.

Communications Performance Highlights for V5R1:

- In V5R1 the scalability was significantly improved. Having efficient scaling means that as throughput increases, CPU consumption increases roughly proportionally to throughput.

- 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.

5.1 TCP/IP, Sockets 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. This is based on measurements gathered from two iSeries Model 820 systems. This table may be used to estimate a system's potential transaction rate at a given CPU utilization assuming a particular workload.

Table 5.1. V5R2 iSeries TCP/IP Capacity Planning	
NetPerf Transaction Type:	**Capacity Metric** (transactions/second per CPW)
Request/Response (RR) 1 Byte	23.3
Asym. Connect/Request/Response (ACRR) 8K Bytes	4.5
Large Transfer (Stream) 16K Bytes	12.7

Notes:
- Capacity metrics are provided for nonsecure and each variation of security policy
- Based on measurements with the NetPerf workload using two iSeries 820 models with V5R2
- The table data reflects iSeries as a server (not a client)
- The data reflects Sockets, TCP/IP and 100 Mbps Ethernet.
- If any of these configuration characteristics are changed, performance may differ significantly.
- CPW is the "Relative System Performance Metric" from Appendix C. 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 connection supporting a small packet request/response application (Model 830/2402, 1-byte request/response) and wishes to use about 20% of the overall CPU for the network processing portion, then note the following calculation:

*4200 CPW * 20% * 23.3 transactions/second/CPW = 19,572 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 IBM eServer Workload Estimator, described in Chapter 22, 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.

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

Table 5.2 shows the Capacity Metrics for Gigabit Ethernet and jumbo frame Gigabit Ethernet.   These metrics were measured between two iSeries 820 models using the NetPerf workload. Gigabit jumbo frames allow decreased 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 almost over 50% when using jumbo frames rather than standard frames.

Table 5.2. V5R2 iSeries Gigabit Ethernet Performance		
	Capacity Metric (Transactions/second per CPW	
	Gigabit Ethernet 1496-Byte MTU	**Gigabit Ethernet 8996-Byte MTU (Jumbo Frame)**
Large Transfer (Stream) 16K Bytes 1 user on 1 adapter	8.3	12.2

Notes:
- Capacity metrics are provided for nonsecure and each variation of security policy
- Based on measurements with the NetPerf workload using two iSeries 820 models with V5R2
- The table data reflects iSeries as a server (not a client)
- The data reflects Sockets, TCP/IP and Gigabit Ethernet.
- If any of these configuration characteristics are changed, performance may differ significantly.
- CPW is the "Relative System Performance Metric" from Appendix C. 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

Performance Observations/Tips

- For a large transactoin environment, Gigabit Ethernet provides over 6 times more throughput than 100 Mbps Ethernet for a single user. If jumbo frames are used, then Gigabit Ethernet provides almost 10 times more throughput than 100 Mbps Ethernet for a single user.

- The gigabit jumbo frame option requires 8996 Byte MTU support by all of the gigabit network components including switches, routers and bridges. For iSeries 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 of data, maximize the size of the application's send and receive requests. 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 the default setting of 8K bytes to 64K bytes to maximize data rates. When transferring large amounts of data with V4R4 , V4R5, V5R1 or V5R2, you may experience higher throughput by increasing these buffer sizes up to 8MB. The exact buffer size that provides the best throughput will be dependent on several network environment factors including types of switches and systems, ack timing, error rate and network topology. For Tables 5.1 and 5.2, 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 after V4R4, so that TCP transactions could always have the higher-level of performance.

- In V4R5 and later 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 (such as 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.

- 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 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.

5.3 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) (prior to V5R1): 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 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).

- 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 average utilization is calculated 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 average
    utilization is 50%, the maximum utilization is 100%.

- 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.

5.4 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:

    - 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)
    - 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).
    - Reducing the pacing values for the large transfer will also slow it down, allowing the interactive users more windows for getting on the line.

5.5 NetPerf Workload Description

The NetPerf workload is a primitive-level function workload used to explore communications performance. The NetPerf workload consists of C programs that run between a client iSeries and a server iSeries. 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).

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 WebSphere Performance

This section discusses iSeries performance information in web serving and WebSphere environments. Specific products that are discussed include: HTTP Server (powered by Apache) (in section 6.1), WebSphere Application Server and WebSphere Application Server - Express (6.2), Web Facing (6.3), WebSphere Portal Server (6.4), WebSphere Commerce (6.5), WebSphere Commerce Payments (6.6), and Connect for iSeries (6.7).

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

Web Overview: There are many factors that can impact overall performance (e.g., end-user response time, throughput) in the complex web environment, some of which are listed below:

1) Web Browser or client
- processing speed of the client system
- performance characteristics and configuration of the Web browser
- client application performance characteristics

2) Network
- speed of the communications links
- capacity and caching characteristics of any proxy servers
- the responsiveness of any other related remote servers (e.g., payment gateways)
- congestion of network resources

3) iSeries Web Server and Applications
- iSeries processor capacity (indicated by the CPW value)
- utilization of key iSeries resources (CPU, IOP, memory, disk)
- web server performance characteristics
- application (e.g., CGI, servlet) performance characteristics

Comparing traditional communications to web-based transactions: For commercial applications, data accesses across the Internet 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
- with the Internet, the network may not be well-understood (route, components, performance)

Information source and disclaimer: The information in the sections that follow is based on performance measurements and analysis done in the internal IBM performance lab. The raw data is not provided here, but the highlights, general conclusions, and recommendations are included. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here. Note that these workloads are measured in best-case environments (e.g., local LAN, large MTU sizes, no errors). Real Internet networks typically have higher contention, higher levels of logging and security, MTU size limitations, and intermediate network servers (e.g., proxy, SOCKS); and therefore, it would likely consume more resources.

6.1 HTTP Server (powered by Apache)

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. FRCA (Fast Response Caching Accelerator) can also be configured to cache the file deeper in the operating system and further reduce resource consumption.
- **CGI**: HTTP invokes a CGI program which builds a simple HTML page and serves it via the HTTP server. This CGI program can run in either a new or a named activation group. The CGI programs were compiled using a "named" activation group unless specified otherwise.
- **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.
- **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 eServer Workload Estimator to do capacity planning for web environments using the following workloads: Web Serving, WebSphere, WebFacing, WebSphere Portal Server, WebSphere Commerce. This tool allows you to suggest a transaction rate and to further characterize your workload. You'll find the tool along with good help text at:

http://www.ibm.com/eserver/iseries/support/estimator . Work with your marketing representative to utilize this tool (also chapter 22).

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 V5R2 iSeries Web Serving Capacity Planning - Various Transaction Types				
	Nonsecure		Secure	
Transaction Type:	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 (IFS)	1.75	0.57	1.47	0.67
Static Page (cache)	2.79	0.35	2.23	0.44
Static Page (FRCA)	13.01	0.08	na	na
CGI (new activation)	0.06	16.12	0.06	16.37
CGI (named activation)	0.37	2.71	0.36	2.73
Persistent CGI	0.28	3.52	0.26	3.85
Net.Data	0.17	5.89	0.17	5.95
User Module	3.15	0.31	2.69	0.37

Notes/Disclaimers:
- IBM HTTP Server (powered by Apache) for iSeries; V5R2; 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, KeepAlive on, 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 C
- Web server capacities may not necessarily scale exactly by CPW, actual results may differ significantly
- Transactions using more complex programs or serving larger files will have lower capacities that what is listed here.
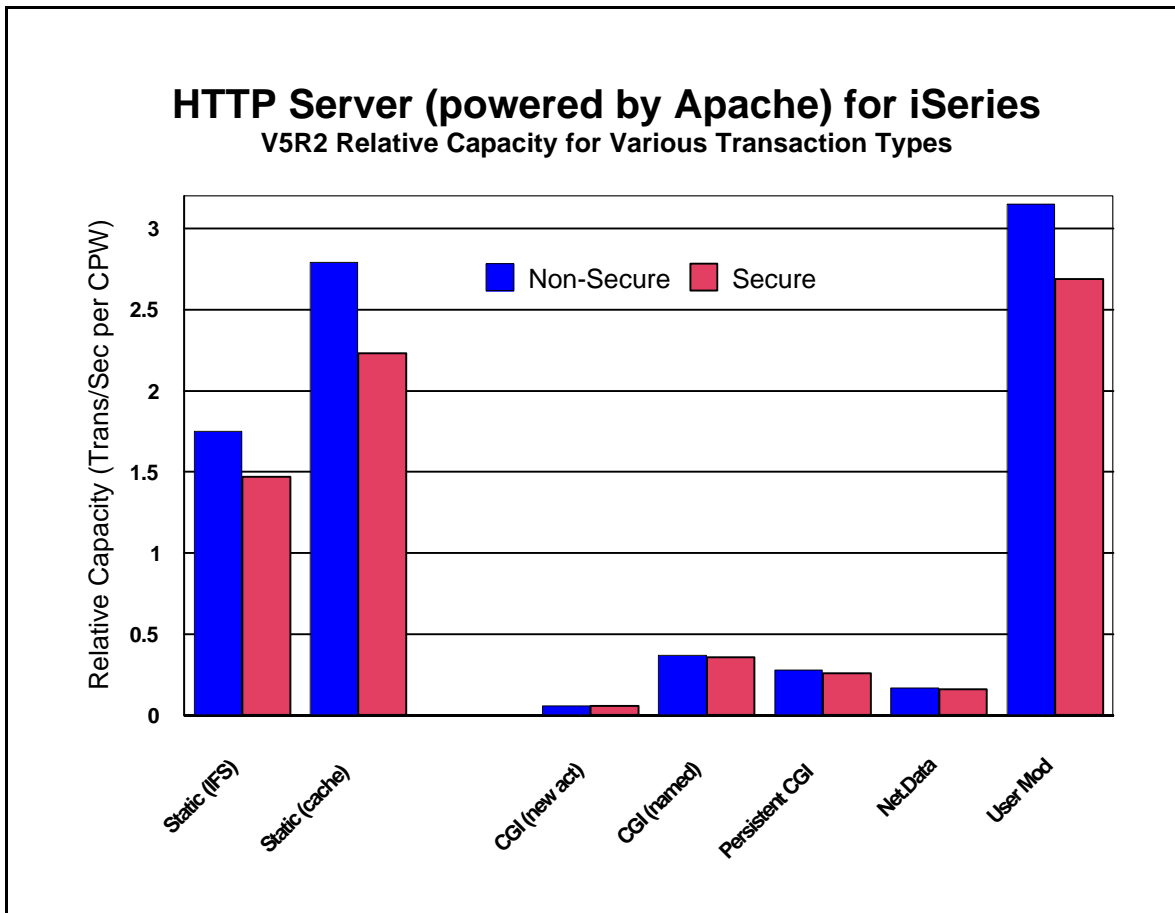
Figure 6.1 iSeries Web Serving V5R2 Relative Capacities - Various Transactions

Table 6.2 V5R2 Relative Capacity for Static (varied sizes)			
	Capacity Metrics Pairs of values (KeepAlive off / KeepAlive on)		
Transaction Type:	**1K Bytes** **(trans/sec per CPW)**	**10K Bytes** **(trans/sec per CPW)**	**100K Bytes** **(trans/sec per CPW)**
Static Page (IFS)	1.24 / 1.75	1.07 / 1.48	0.48 / 0.57
Static Page (local cache)	1.72 / 2.79	1.46 / 2.25	0.49 / 0.58
Static Page (FRCA)	4.97 / 13.01	3.51 / 6.21	0.83 / 1.03
Notes/Disclaimers: • IBM HTTP Server (powered by Apache) for iSeries; V5R2; 100Mbps Ethernet • Based on measurements from an iSeries Model 270 • CPWs are "Relative System Performance Metrics" listed in Appendix C • 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			
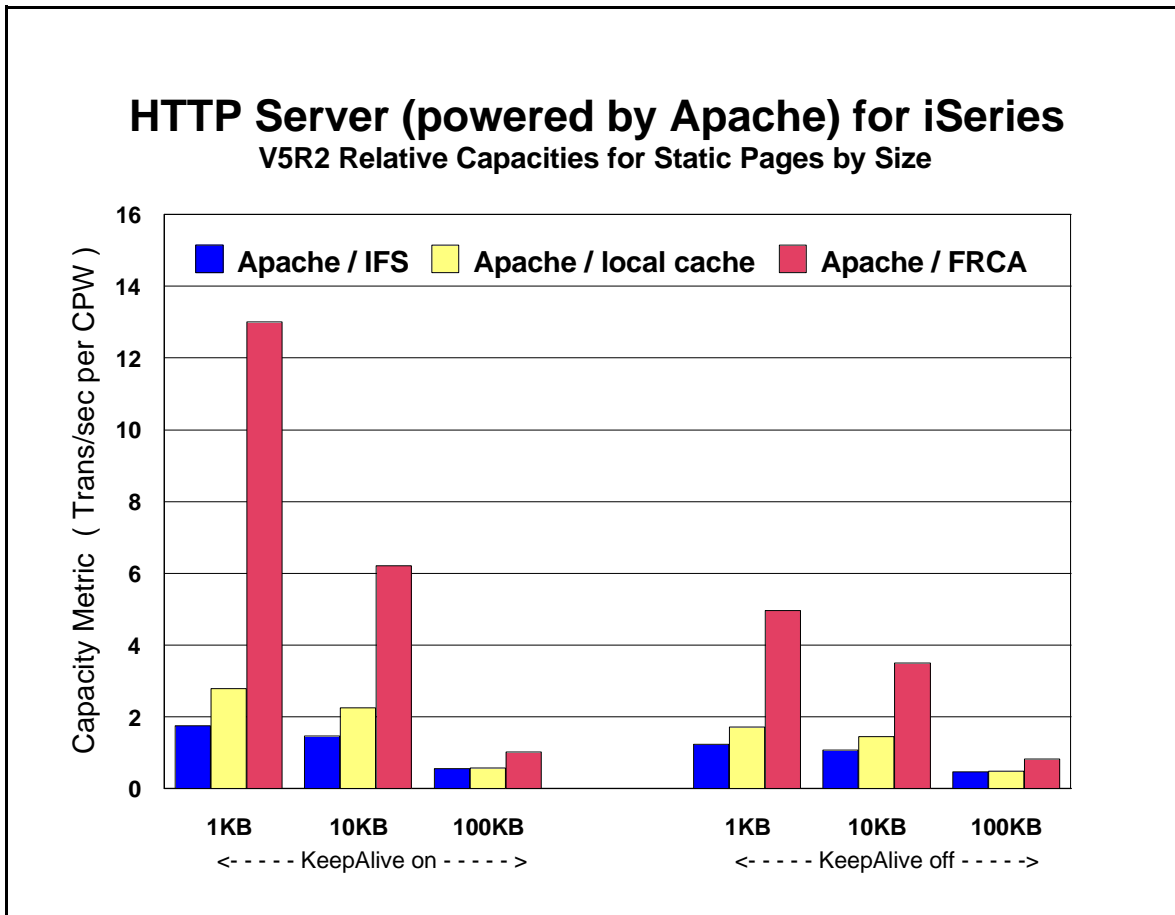
Figure 6.2 V5R2 Relative Capacity for Static Pages and FRCA

Web Serving Performance Tips and Techniques:

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

   a. **V5R2** provides opportunities to exploit improved performance. HTTP Server (powered by Apache) was updated to current levels with improved performance and scalability. FRCA (Fast Response Caching Accelerator) is new with V5R2 and provides a high-performance compliment to the HTTP Server for highly-used static content. FRCA generally reduces the CPU consumption to serve static pages by half, potentially doubling the web server capacity.

   b. **V5R1** provided the new HTTP Server (powered by Apache) and continues to support the HTTP Server (original). Note that the performance information provided in the tables represents HTTP Server (powered by Apache). In general, the performance of the Apache server is equal to or better than the original server. *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*

   c. **V4R5** provided 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.

   d. **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.

   e. **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, Tables 6.1 and 6.2 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 2054 trans/sec (1070 CPWs * 1.75 trans/sec/CPW = 1872 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 410 trans/sec (1070 CPWs * 1.75 trans/sec/CPW * 20% = 374 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, 429 CPWs would be required (825 trans/sec * 0.57 CPW/trans/sec = 470 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 locally in the HTTP Server's cache can significantly increase web server capacity (refer to Table 6.2 and Figure 6.2). Ensure that highly used files are selected to be in the cache to limit the overhead of accessing IFS. To keep the cache most useful, it may be best not to consume the cache with extremely large files. Ensure that highly used small/medium files are cached. Also, consider using the LiveLocalCache off directive if possible. If the files you are caching do not change, you can avoid the processing associated with checking each file for any updates to the data. A great deal of caution is recommended before enabling this directive**.**

4. **FRCA:** Fast Response Caching Accelerator is newly implemented for V5R2. FRCA is based on AFPA (Adaptive Fast Path Architecture), utilizes NFC (Network File Cache) to cache files, and interacts closely with the HTTP Server (powered by Apache). FRCA greatly improves web server performance for serving static content (refer to Table 6.2 and Figure 6.2). For best performance, FRCA should be used to store static, nonsecure content (pages, gifs, images, thumbnails). Keep in mind that HTTP requests served by FRCA are not authenticated. Taking advantage of all levels of caching is really the key for good e-Commerce performance (local HTTP cache, FRCA cache, WebSphere Commerce cache, etc.).

5. **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 eServer Workload Estimator can be used for capacity planning with page size variations (see chapter 22).

6. **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.

7. **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 Figure 6.1 that the performance of CGI (using named activations) is nearly identical to that of persistent CGI.

8. **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.

9. **Apache User Modules:** The HTTP Server (powered by Apache) provides support for user modules. These highly flexible user-written programs 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.).

10. **Secure Web Serving**: Secure web serving involves additional overhead to the server for web environments. There are primarily two groups of overhead: First, there is the fixed overhead of establishing/closing a secure connection, which is dominated by key processing. Second, there is the variable overhead of encryption/decryption, which is proportional to the number of bytes in the transaction. Note the capacity factors in the tables above comparing nonsecure and secure serving. From Table 6.1, note that simple transactions (e.g., static page serving), the impact of secure serving is around 20%. For complex transactions (e.g., CGI, Net.Data, servlets), the overhead is more watered down. This relationship assumes that KeepAlive is used, and therefore the overhead of key processing can be minimized. If KeepAlive is not used (i.e., a new connection, a new cached or abbreviated handshake, more key processing, etc.), then there will be a hit of 3x or more CPU time for using secure transaction. To illustrate this, a noncached SSL static transaction using KeepAlive is 1.47 trans/sec per CPW (from Table 6.1); this compares to 0.52 trans/sec per CPW (not included in the table) when KeepAlive is off. However, if the handshake is forced to be a regular or full handshake, then the CPU time hit will be around 50x (0.03 trans/sec per CPW). The lesson here is to: 1) limit the use of security to where it is needed, and 2) use KeepAlive if possible.

11. **Persistent Requests and KeepAlive**: Keeping the TCP/IP connection active during a series of transactions is called persistent connection. Taking advantage of the persistent connection for a series of web transactions is called Persistent Requests or KeepAlive. This is tuned to satisfy an entire typical web page being able to serve all imbedded files on that same connection.

     a. Performance Advantages: The CPU and network overhead of establishing and closing a connection is very significant, especially for secure transactions. Utilizing the same connection for several transactions usually allows for significantly better performance, in terms of reduced resource consumption, higher potential capacity, and lower response time.

     b. The down side: If persistent requests are used, the web server thread associated with that series of requests is tied up (only if the Web Server directive AsyncIO is turned Off). If there is a shortage of available threads, some clients may wait for a thread non-proportionally long. A time-out

parameter is used to enforce a maximum amount of time that the connection and thread can remain active.

12. **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.

13. **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.

14. **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.

15. **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".

16. **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.**.**

17. **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.

18. **Communications/LAN IOPs**: Since there are a dozen or more line flows per transaction (assuming KeepAlive is off), the Web serving environment utilizes the IOP more than other communications environments.   Use the Performance Monitor or Collection Services to measure IOP utilization. Attempt to keep the average IOP utilization at 60% or less for best performance.   IOP capacity depends on page size, the MTU size, the use of Keep Alive, etc. For the best projection of IOP capacity, consider a measurement and observe the IOP utilization.

6.2 WebSphere Application Server

This section discusses iSeries performance information for the WebSphere Application Server. This section will address the following packages: WebSphere Application Server V4.0 Advanced Edition, WebSphere Application Server V4.0 Advanced Edition Single Server, WebSphere Application Server V5, and WebSphere Application Server Express V5 performance. Historically, both WebSphere and OS/400

Java performance continues to improve with each version. Note from the figures and data in this section that the most recent versions of WebSphere and/or OS/400 provides the best performance.

Tuning for WebSphere is important to achieve optimal performance. Please refer to the *WebSphere Application Server for iSeries Performance Considerations* documents for more information. These documents describe the performance differences between the different WebSphere Application Server versions on iSeries. They also contain many performance recommendations for environments using servlets, Java Server Pages (JSPs), and Enterprise Java Beans. To access these documents use the following URL:

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

Although some capacity planning information is included in these documents, please use the eServer Workload Estimator as the primary tool to size WebSphere environments. The Workload Estimator is kept up to date with the latest capacity planning information available.

Trade2 Benchmark (WebSphere eBusiness Benchmark) Description:

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".

Trade2 Application Runtime Topology

Trade2

http/htttps

HTTP Server

request

response

Web Browser

WebSphere Application Server

Servlets

JSPs

Business Data Access

EJB

JDBC

Enterprise Data
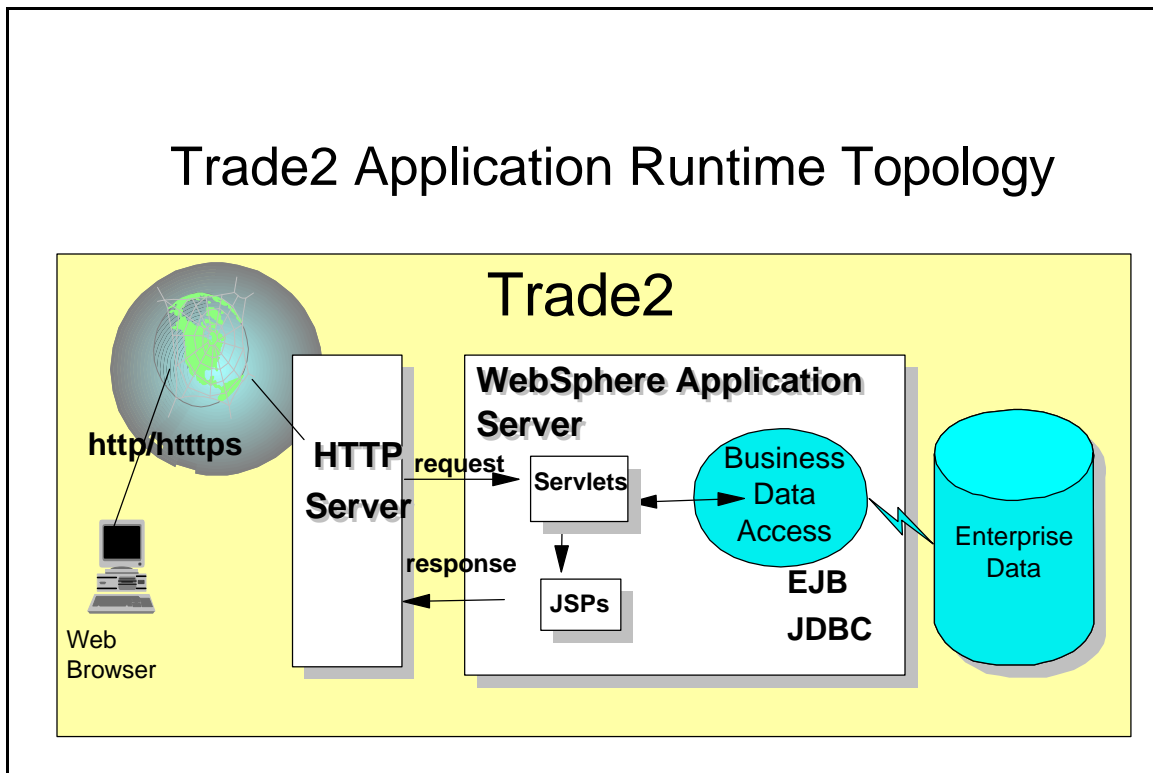
Figure 6.3 Topology of theTrade Application

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.

Trade2 Measurement Results:

Trade2.5 on iSeries - Historical View
Capacity - model 170/2385

(Chart: Transactions/second, y-axis 0 to 120)

Categories: TradeJDBC, TradeEJB

Legend:
- V4R5 WAS3.0.2
- V4R5 WAS3.5.0
- V4R5 WAS3.5.1
- V4R5 WAS3.5.2
- V4R5 WAS3.5.3
- V5R1 WAS3.5.3
- V5R1 WAS4.0.1
- V5R2 WAS4.0.3*
- V5R2 WAS 5.0*
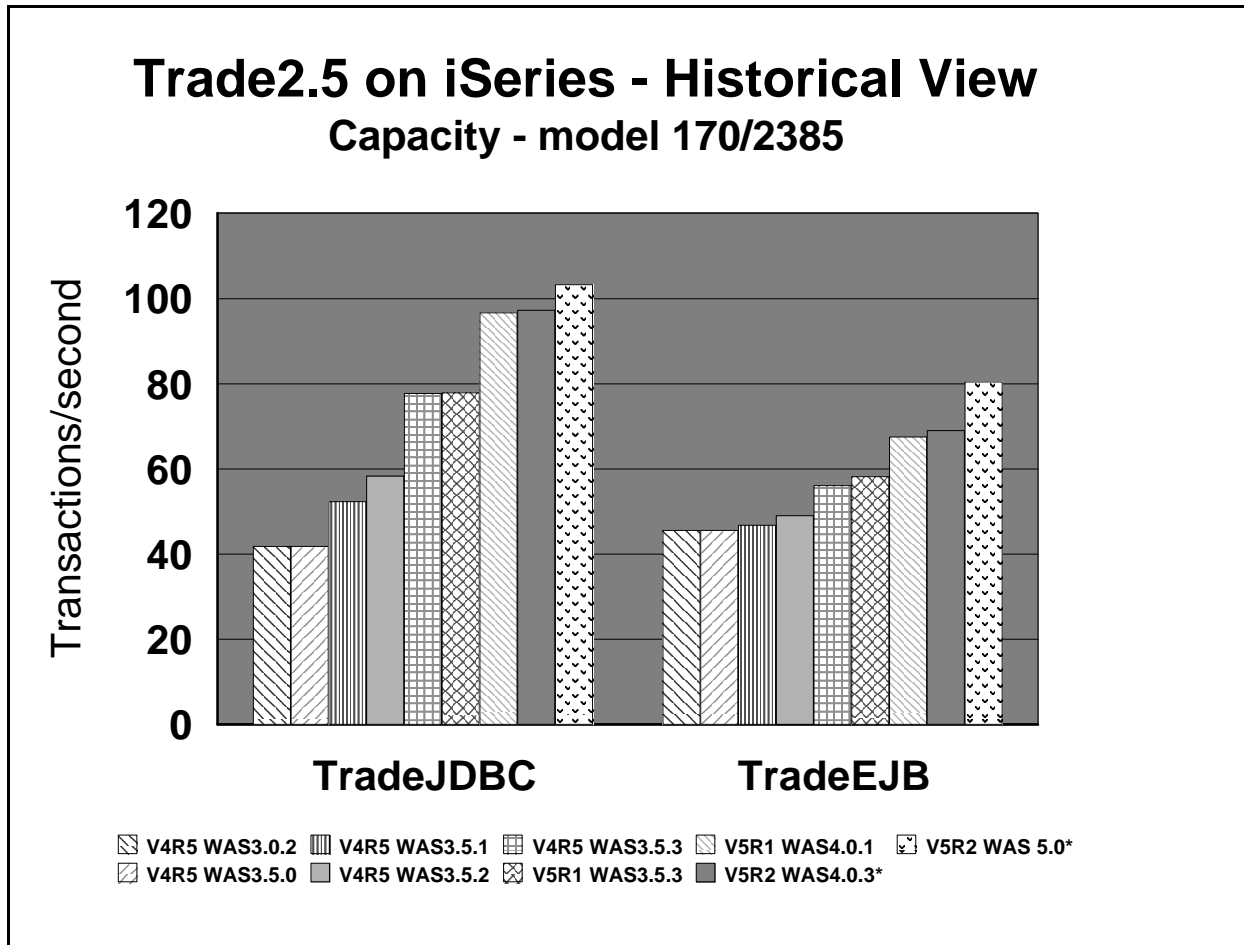
Figure 6.4 Trade2 1-way Results

WebSphere Application Server Trade2 Results
Notes/Disclaimers:
• Results were measured on a 170/2385 system • Trade2 JDBC and Trade2 EJB benchmarks • WebSphere 3.0.2, 3.5.0, 3.5.1, and 3.5.2 were on a V4R5 system • WebSphere 3.5.3 was measured on both V4R5 and V5R1 • WebSphere 4.0 AE was measured on V5R1 • WebSphere 4.0.3 AE on V5R2 was estimated via measurements with WAS 4.0.2 with software enhancements to be included with WAS 4.0.3 • The IBM HTTP Server (powered by Apache) was used starting with the V5R2 measurements • * - Results are projected from Trade2.7, which is 20% heavier then Trade 2.5

Trade2 Scalability Results

iSeries Trade2
High Water Scaling of Hardware and Software

(Bar chart: Transactions/Second vs. EJB and JDBC)

Legend:
- V4R5 - WAS 3.0.2
- V5R1 - WAS 4.0
- V5R2 - WAS 5.0
- V5R1 - WAS 3.5.3
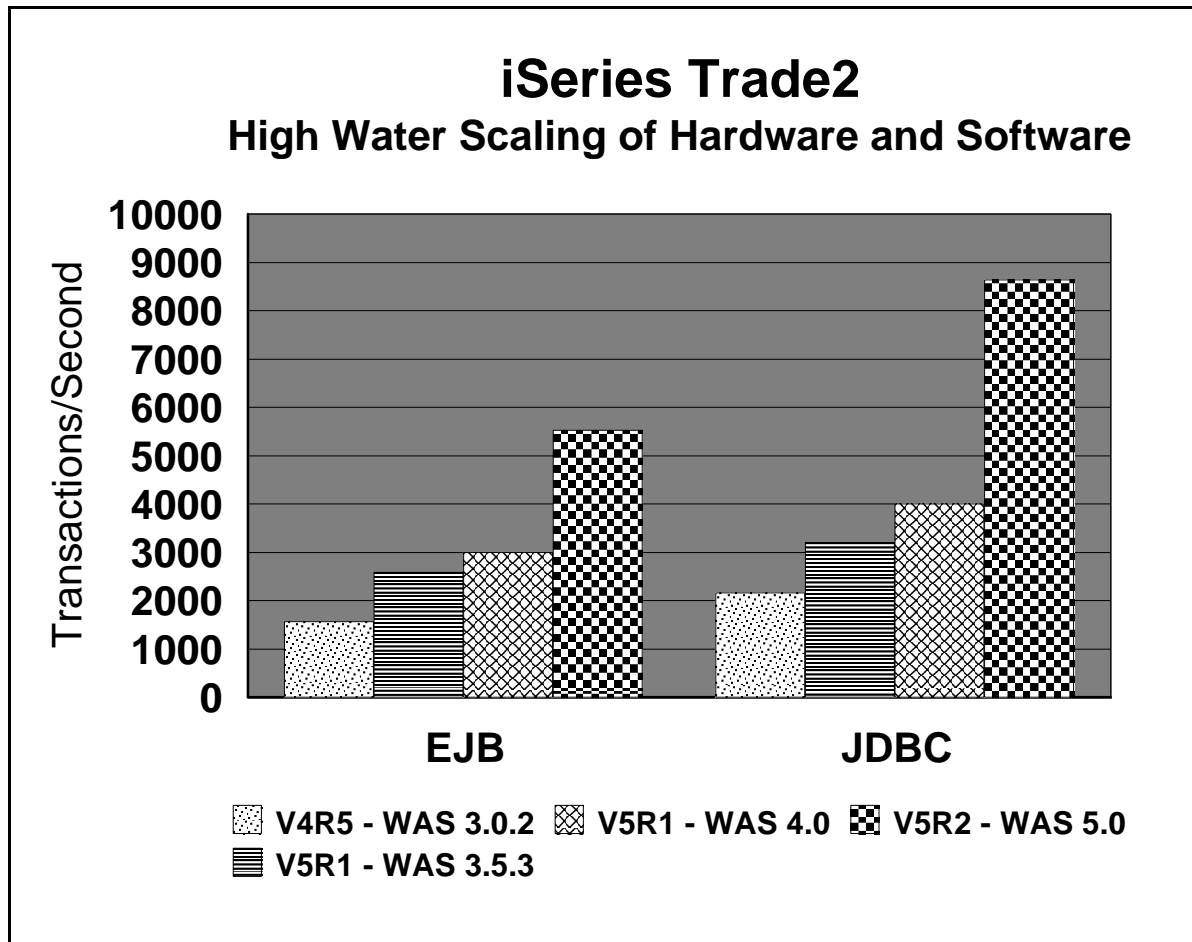
Figure 6.5 Trade2 24-way Results

WebSphere Application Server Trade2 Results
Notes/Disclaimers:
• Results were measured on either a 24-way system or 32-way system. As a general rule of thumb, we see approximately a 10% improvement using WAS 5.0, compared to WAS 4.0. Please see the Primatives section for a better comparison of WAS 4.0 and 5.0, on the same hardware.
• V4R5 - 840/2420 24-Way 500 MHz
• V5R1 - 840/2461 24-Way 600 MHz
• V5R2 - 890/2488 32-Way 1.3 GHz
• Trade2 JDBC and Trade2 EJB benchmarks
• V4R5 was measured with WebSphere 3.0.2
• V5R1 was measured with WebSphere 3.5.3 and WebSphere 4.0 AE
• V5R2 was measured with WebSphere 5.0
• IBM HTTP Server

Trade2 Primitives

Trade provides a suite of web primitives, which singularly test key operations in the enterprise Java programming model. These primitives are very useful in the Rochester lab for release-to-release comparison tests, to determine if a degradation occurs between releases, and what area to focus

performance improvements. Table 6.3 describes all of the primitives that are shipped with
Trade2, and Figure 6.6 shows the results of the primitives from WAS 4.0 and WAS 5.0.

Primitive Name	Description of Primitive
PingHtml	The most basic operation providing access to a simple "Hello World" page of static HTML.
PingServlet	Fundamental dynamic HTML creation through server side servlet processing.
PingServletWriter	Extends PingServlet by using a PrintWriter for formatted output vs. the output stream used by PingServlet.
PingServlet2Servlet	Tests request dispatching. Servlet 1, the controller, creates a new JavaBean object forwards the request with the JavaBean added to Servlet 2. Servlet 2 obtains access to the JavaBean through the Servlet request object and provides dynamic HTML output based on the JavaBean data.
PingJSP	Tests a direct call to JavaServer Page providing server-side dynamic HTML through JSP scripting.
PingServlet2JSP	Tests a commonly used design pattern, where a request is issued to servlet providing server side control processing. The servlet creates a JavaBean object with dynamically set attributes and forwards the bean to the JSP through a RequestDispatcher The JSP obtains access to the JavaBean and provides formatted display with dynamic HTML output based on the JavaBean data.
PingHTTPSession1	Session1 tests a fundamental HTTP session function by creating a unique session ID for each individual user. The ID is stored in the users session and is accessed and displayed on each user request.
PingHTTPSession2	Session2 extends Session1 by invalidating the HTTP Session on every 5th user access. This results in testing HTTPSession create and destroy
PingHTTPSession3	Tests the servers ability to manage and persist large HTTPSession data objects. The servlet creates a large custom java object. The class contains multiple data fields and results in 2048 bytes of data. This large session object is retrieved and stored to the session on each user request.
PingJDBCRead	Tests fundamental servlet to JDBC access to a database performing a single-row read
PingJDBCWrite	Tests fundamental servlet to JDBC access to a database performing a single-row write
PingJDBCReadPrepStmt	Extends the basic JDBC test by using a prepared statement for database read access. Statement parameters are set dynamically on each request.
PingJDBCWritePrepStmt	Extends the basic JDBC test by using a prepared statement for database write access. Statement parameters are set dynamically on each request.
PingServlet2SessionEJB	Tests a key function of a servlet call to a stateless SessionEJB. The SessionEJB performs a simple calculation and returns the result
PingServlet2EntityEJB	Tests key function of a servlet call to a Container Managed persistent Entity EJB. In this test the EJB entity represents a single row in the database table. The values are accessed through the EJB and displayed
PingServlet2EntityEJB2	This test extends the previous EJB Entity test by using a finder method on the Entity that returns a collection of Entity objects. Each object is displayed by the servlet
PingServlet2Session2Entity	This tests the full servlet to Session EJB to Entity EJB path to retrieve a single row from the database.

Table 6.3 Description of Trade primitives in Figure 6.6

WebSphere Application Server Trade2 primitives Release to Release comparison

Chart legend:
- WAS 4.0.4
- WAS 5.0

Y-axis: Transactions Per CPW (0, 0.5, 1, 1.5, 2)

X-axis categories: PingHtml, PingServlet, PingServletWriter, PingServlet2Servlet, PingJSP, PingServlet2JSP, PingHTTPSession1, PingHTTPSession2, PingHTTPSession3, PingJDBCRead, PingJDBCWrite, PingJDBCReadPrepStmt, PingJDBCWritePrepStmt, PingServlet2SessionEJB, PingServlet2EntityEJB, PingServlet2EntityEJB2, PingServlet2Session2Entit

Figure 6.6 WebSphere Trade2 primitive results. The measurements were performed on the same machine, an 820-2397 500 MHz 2-Way. The results shown are transaction per CPW. All results are for a non-secure environment.

WebSphere Application Server V5 Express

Starting with V5 WebSphere has a new offering, called WebSphere Application Server Express. Express is an attractive solution for many small and midsized customers. Whether the customer is deploying an application which requires a WebSphere solution (such as WebFacing), or the customer is looking to write their own solution to leverage the Web, Express can be a lower cost alternative to the full J2EE compliant WebSphere Application Server V5. There are features that the base WebSphere Application Server has that Express does not have. The most notable of these features include EJB support, and horizontal and vertical scaling using clones. Because WebSphere Express does not support the EJB version, all of the measurements in the Express section are running the JDBC version of Trade2. Underneath the covers, the WebSphere Express code base is almost equivalent to the WebSphere Application Server Base. The only differences are the added features of WAS Base, and the way WAS Express is packaged (See below).

The biggest concern from small machine customers in the past has been the startup time of the application server. Keeping this in mind, Express is specifically tuned for machines with lower CPW ratings. In the base WebSphere Application Server V5, as the application server starts up, numerous WebSphere classes

Trade2 Startup Times, Comparison Between WAS Express and Base

01:57

01:38

02:18
02:01
01:44
01:26
01:09
00:52
00:35
00:17
00:00

☐ WAS Express (300 CPW)
☐ WAS Base (300 CPW)
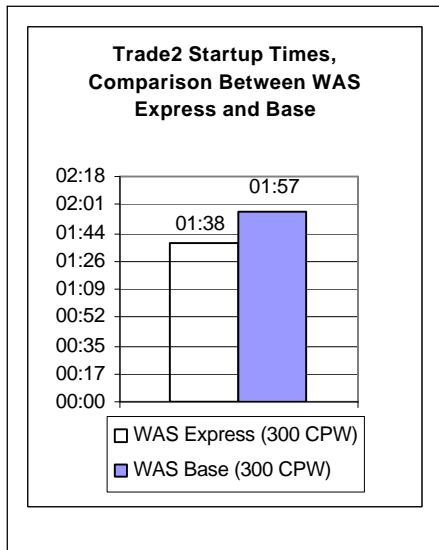
Figure 6.7 - WebSphere Application Server V5 Express starts up over 16% faster then the WebSphere V5 Base, running the Trade2 Benchmark

get loaded into the JVM, which can either cause the Java class to run in interpret mode using Mixed Mode Interpreter (MMI)or run the Just In Time (JIT) Java compiler against the class, which will compile the class during runtime. Both of these options are CPU intensive, and lower CPW machines do not handle them well. Express gets around this by shipping the Direct Execution (DE) Java programs for the classes that are used during startup. These classes are loaded and executed the first time more quickly, due to the fact that the high performing DE java programs have already been compiled. The result is shown in Figure 6.7. The Trade2 application server starts up over 16% faster with Express, when compared to base. These measurements were done on the Model 800, a 300 CPW machine, which is the IBM recommended minimum for WebSphere Express. When going to higher CPW machines, this difference will become smaller and smaller, due to the increased CPU available.

The drawback to shipping the DE programs with WAS Express is two fold. First and foremost, is that the JIT compiler has recently become faster during runtime then DE, due to new technology. Thus, once the application is up and running, you can expect slightly worse performance with Express, than with base WebSphere Application Server V5. Figure 6.8 shows that the performance of WebSphere Express delivers 11% less throughput than WebSphere V5 Base while running the Trade2 application. This is due to the fact that the JIT compiled code generated on the fly can perform more optimizations then the static code generated by DE. It is important to note that your application will likely not see the same 11% degradation that Trade2 experienced, but will experience some degradation. Secondly, the amount of

disk space needed for Express and the installation CDs will be larger than WebSphere V5 Base, due to the DE programs being stored on the system and the installation CDs. For more information about the MMI, JIT and DE performance, please see Chapter 7 - Java.

Although IBM and the Workload Estimator do not recommend any machine lower then the 300 CPW machine to run WebSphere Application Server V5 Express, additional tests have been performed against a Model 270-2248, which is a 150 CPW machine. These tests, in Figure 6.9, proved that it is possible to run Express on this machine, with cautious planning. When doing this, however, the customer must be aware of the longer startup times and decreased throughput, and be sure they are tolerable in their environment.

Trade2 Runtime Throughput, Comparison Between WAS Express and Base

67.00

59.75

70.00
60.00
50.00
40.00
30.00
20.00
10.00
0.00

Trans / Sec

☐ WAS Express 300 CPW
☐ WAS Base (300 CPW)

Figure 6.8 WebSphere Application Server V5 Express is about 11% slower then WebSphere V5 Base, running the Trade2 Benchmark.

Use the eServer Workload Estimator to predict the capacity characteristics for WebSphere Application Server V5 Express performance (using the WebSphere workload category). The Workload Estimator contains the most recent capacity planning data available, and will ask you a series of questions that make sense to your application. You can find the tool at: http://www.ibm.com/eserver/iseries/support/estimator.   A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (see also chapter 22).

Trade Startup Times

05:14

WAS Express (150 CPW)

Trade2 Runtime Throughput

29.67

WAS Express 150 CPW

Figure 6.9 WebSphere Application Server V5 Express results on Model i270-2248, 150 CPW machine. With careful planning and low expectations, a smaller customer can see acceptable results with the 150 CPW machine running WAS Express, even though it is not a recommended system from IBM

6.3 IBM WebFacing

The IBM WebFacing tool converts your 5250 application DDS display files, menu source, and help files into Java Servlets, JSPs, JavaBeans, and JavaScript to allow your application to run in either WebSphere Application Server V5 or V4. This is an easy way to bring your application to either the Internet, or the Intranet, both quickly and inexpensively.

The Number of Input/Output fields are the main metric to tell how heavy a WebFaced application will be on the WebSphere Application Server. The number of Input/Output fields are simple to count for most of the screens, except when dealing with subfiles. Subfiles can affect the number of input/output fields dramatically. The number of fields in subfiles are significantly impacted by two DDS keywords:
1. SFLPAG - The number of rows shown on a 5250 display.
2. SFLSIZ - The number of rows of data extracted from the database.

When using a DDS subfile, there are 3 typical modes of operation:
1. SFLPAG=SFLSIZ. In this mode, there are no records that are cached. When more records are requested, WebFacing will have to get more rows of data. This is the recommended way to run your WebFacing application.
2. SFLPAG < SFLSIZ. In this mode, WebFacing will get SFLSIZ rows of data at a time. WebFacing will display SFLPAG rows, and cache the rest of the rows. When the user requests more rows with a page-down, WebFacing will not have to access the database again, unless they page below the value of SFLSIZ. When this happens, WebFacing will go back to the database and receive more rows.
3. SFLPAG = (SFLSIZ) * (Number of times requesting the data). This is a special case of option 2 above, and is the recommended approach to run GreenScreen applications. For the first time the page is requested, SFLPAG rows will be returned. If the user

Average Number of Fields per Panel

Constant Overhead Per Panel □ Overhead for I/O Fields
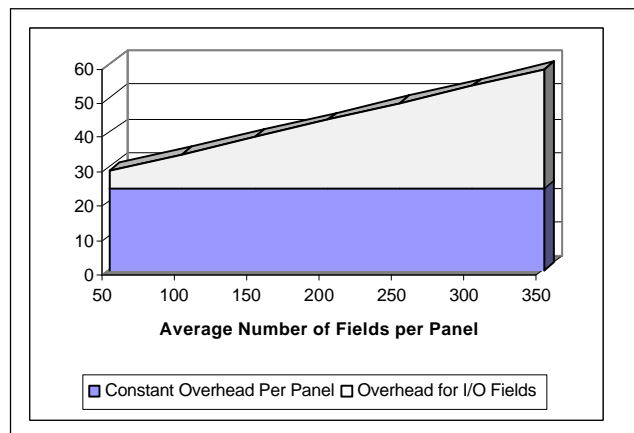
Figure 6.10 Shows the impact on CPU that the number of I/O fields has per WebFaced panel

V5R2 Performance Capabilities Reference - January 2003

performs a page down, then SFLPAG * 2 rows will be returned. This is very efficient in 5250 applications, but less efficient with WebFacing.

Since WebFacing is performance sensitive to the number of input/output fields that are requested from WebFacing, the best option would be the first mode, since this will minimize the number of these fields for each 5250 panel requested through WebFacing. The number of fields for a subfile is the number of rows requested from the database (SFLSIZE) times the number of columns in each row.

Figure 6.10 shows a theoretical algorithm to graphically describe the effect the number of Input/Output fields has on the performance of the WebFaced application. The Y-axis metric is not important, but merely can be used to measure the relative amount of CPU horsepower that the application needs to serve one single 5250 panel. In this case, serving one single panel with 50 I/O fields is approximately one half the CPU horsepower needed to serve one 5250 panel with 350 I/O fields. As you can see, the number of I/O fields dramatically impacts the performance of your WebFacing application, thereby reducing the I/O fields will improve your performance.

Name	Average number of I/O Fields / panel
Workload A	37
Workload B	99
Workload C	612

Table 6.4 Average number of I/O fields for each workload defined in this section.

Model	CPW
800-2463	300
i270-2388	1,090
i810-2469	2,700

Table 6.5 iSeries models used in WebFacing studies conducted in the Rochester lab.

In our studies, we selected three customer WebFaced applications, one simple, one moderate, and one complex. See table 6.4, for details on the number of I/O fields for each of these workloads. We ran the workloads on three separate machines (see table 6.5) to validate the performance characteristics with regard to CPW. In our running of the workloads, we tolerated only a 1.5 second **server response time** per panel. This value does not include the time it takes to render the image on the client system, but only the time it took the server to get the information to the client system. The machines that we used are in Table 6.5, and include the 800 and i810 (V5R2 Hardware) and the 170 (V4R4 Hardware). All systems were running OS/400 V5R2.

Some of the results that we saw in our tests are shown in Figure 6.11. This figure shows the scalability across different hardware running the same workload. A user is defined as a client that requests one new 5250 panel every 15 seconds. According to our tests, we see relatively even results across the three machines. The one machine that is a blip is the V4R4 hardware (1090 CPW). This blip can be explained by release-to-release degradation. Since the CPW measurement was made in V4R4, there have been three major releases, each bringing a slight degradation in performance. Thus, if the model 170 was tested with CPW today with V5R2, it would likely be about 10% less then when tested with V4R4. This would bring the 1090 CPW numbers down 10%, and closer in line with the other two machines.

Figure 6.11 CPW per User across the machines documented in table 6.5

Memory impacts are caused by two factors in WebFacing. The first factor is the throughput of the application. As the application server gets more and more busy, more memory is required to run

WebFacing. The second factor is the number of unique WebFacing panels. As the number of unique panels grows, the amount of memory needed to run the WebFacing application also grows. A rule of thumb that is out there today is that you will need 512MB + (64MB for every 100 unique panels) memory on your system.

Many 5250 applications have been implemented with "best performance" techniques, such as minimized number of fields and amount of data exchanged between the device and the application. Other 5250 applications may not be as effficiently implemented, such as restoring a complete window of data, when it was not required. Therfore it is difficult to give a generalized performance comparison between the same application written to a 5250 device and that application using WebFacing to a browser. In the three workloads that we measured, we saw a significant amount of resource needed to WebFace these applications. The numbers varied from 3x up to 8x the amount of CPU resources needed for the 5250 green screen application.

Use the eServer Workload Estimator to predict the capacity characteristics for IBM WebFacing This site will be updated, more often then this paper, so it will contain the most recent information. The Workload Estimator will ask you to specify a transaction rate (5250 panels per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of the panels and the number of unique panels that are displayed by the JSP.   You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (also see chapter 22).

6.4 WebSphere Portal Server

IBM WebSphere Portal V4.1.3 enables companies to build a portal web site serving the individual needs of their employees, business partners and customers. Users can sign on to the portal and view personalized web pages that provide access to the information, people and applications they need. This personalized, single point of access to resources reduces information overload, accelerates productivity and increases web site usage. As WebSphere Portal V4.1.3 supports access through mobile devices, as well as the desktop browser, critical information is always available.

Use the eServer Workload Estimator to predict the capacity characteristics for WebSphere Portal Server (using the WebSphere Portal Server workload category). The Workload Estimator will ask you questions about your portal pages served, such as the number of portlets per page, the complexity of each portlet, and to specify a transaction rate (visits per hour) for a peak time of day. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (see also chapter 22).

6.5 WebSphere Commerce

Use the eServer Workload Estimator to predict the capacity characteristics for WebSphere Commerce performance (using the Web Commerce workload category). The Workload Estimator will ask you to specify a transaction rate (visits per hour) for a peak time of day. It will further attempt to characterize your workload by considering the complexity of shopping visits (browse/order ratio, number of

transactions per user visit, database size, etc.). Recently, the Estimator has also been enhanced to include WebSphere Commerce Pro Entry Edition. The Web Commerce workload also incorporates WebSphere Commerce Payments to process payment transactions. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator. A workload description along with good help text is available on this site. Work with your marketing representative to utilize this tool (see also chapter 22).

6.6 WebSphere Commerce Payments

Use the eServer Workload Estimator to predict the capacities and resource requirements for WebSphere Commerce Payments. The Estimator allows you to predict a standalone WCP environment or a WCP environment associated with the buy visits from a WebSphere Commerce estimation. Work with your marketing representative to utilize this tool. You'll find the tool at: http://www.ibm.com/eserver/iseries/support/estimator.

Workload Description: The PayGen workload was measured using clients that emulate the payment transaction initiated when Internet users purchase a product from an e-commerce shopping site. The payment transaction includes the Accept and Approve processing for the initiated payment request. WebSphere Commerce Payments has the flexibility and capability to integrate different types of payment cassettes due to the independent architecture. Payment cassettes are the plugins used to accommodate payment requirements on the Internet for merchants who need to accept multiple payment methods. For more information about the various cassettes, follow the link below: http://www-4.ibm.com/software/webservers/commerce/paymentmanager/lib.html

Performance Tips and Techniques:

1. **DTD Path Considerations:** When using the Java Client API Library (CAL), the performance of the WebSphere Commerce Payments can be significantly improved if the merchant application specifies the dtdPath parameter when creating a PaymentServerClient. When this parameter is specified, the overhead of sending the entire IBMPaymentServer.dtd file with each response is avoided. The dtdPath parameter should contain the path of the locally stored copy of the IBMPaymentServer.dtd file. For the exact location of this file, refer to the *Programmer's Guide and Reference* at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgrprog22as.html

2. **Other Tuning Tips:** More performance tuning tips can be found in the *Administrator's Guide* under Appendix D at the following link: http://www-4.ibm.com/software/webservers/commerce/payment/docs/paymgradmin22as.html

3. **WebSphere Tuning Tips:** Please refer to the WebSphere section in section 6.2, for a discussion on WebSphere Application Server performance as well as related web links**.**

6.7 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
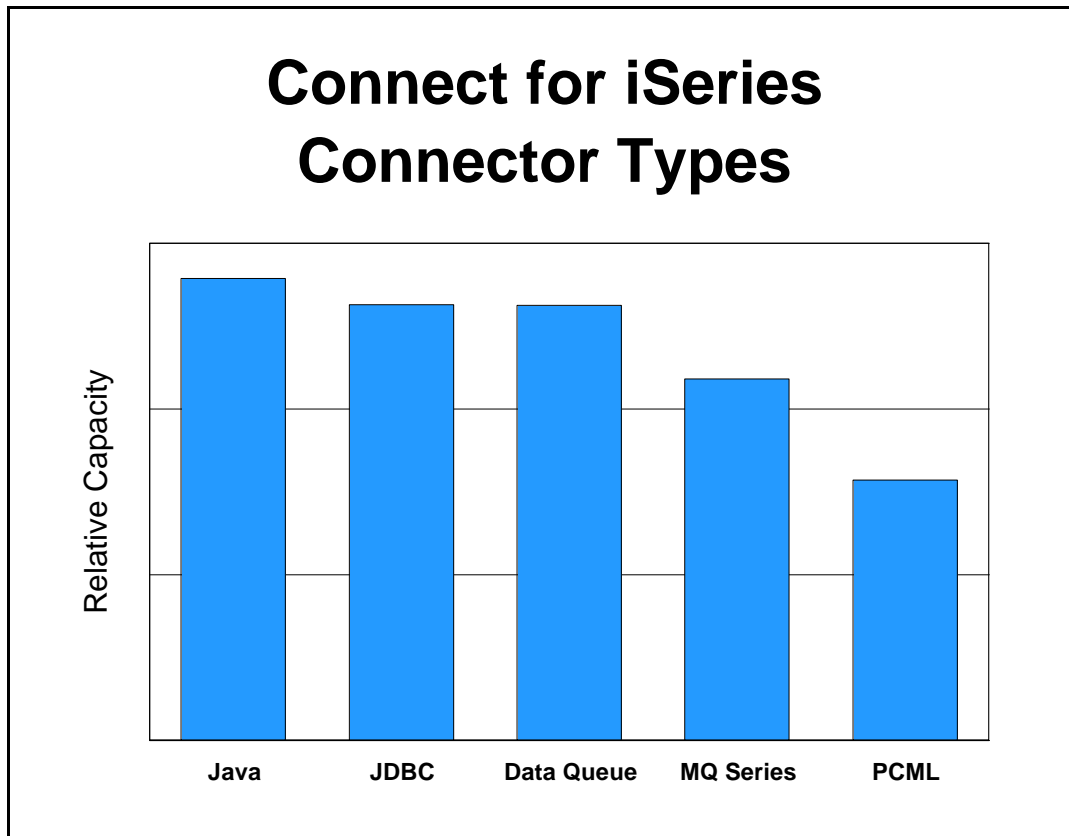
Figure 6.12 Connect for iSeries - Connector Types

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. Java Performance

Highlights:

- Introduction
- Hardware Improvements in V5R2
- Just In Time Compilation in Java
- 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. It is the performance 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 occasionally 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 and improvements to the JIT in V5R2 have improved performance even further, by 15% or more.

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 the WebSphere Application Server and the WebSphere Suite of Applications simply require Java and other advanced function, like XML, will find Java the most fitting choice.

In fact, the world of the web, servlets, J2EE, and the emerging e-business function in general is becoming the premier place to deploy Java with WebSphere 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.

7.2 Hardware Improvements

V5R2 introduces new hardware with performance improvements above those delivered in V5R1.

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, Java could be suitable on these machines. However, this would require a working prototype to suggest precise workload costs. In general WebSphere and other Java environments will tend to produce applications requiring 270, 800, 810, or 820 class machines at a minimum. The eServer 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. In the past OS/400 Java, by contrast, has featured the Transformer technology. The Transformer creates hidden, compiled programs associated with the .class, jar or zip file. The resulting programs are called Direct Execution programs that are fully compliant with the Java standards. In the past, our Java Transformer usually gave better computational performance, at least at the highest optimization levels. However, new enhancements to the JIT compiler in V5R2, and available via a PTF in V5R1, result in better computational performance for JIT than Direct Execution programs.

Also in V5R2 the JIT now has a Mixed Mode Interpreter (MMI). This allows the classes to be loaded without any optimizations then once a method has been executed a set number of times it will then be optimized by the JIT compiler. MMI is enabled by default with the JIT and the default number of times a class will be executed before it is optimized is 2000. This value can be changed by setting the command line property os400.jit.mmi.threshold to equal the desired value. This feature dramatically improves the startup times of a program using JIT.

Note, 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 and the JIT is invoked instead.

There are several advantages to the JIT over Direct Execution besides better runtime performance. The JIT can generate model specific code, so the program is optimized for the hardware it is running on. Also using the JIT will result in dramatically smaller program sizes since there is no hidden Direct Execution Program needed. Classes, jars, or zips are also easier to maintain since no CRTJVAPGM needs to be done when a change is made to a class file. This also saves time, depending on the class, .jar, or .zip file size, since a CRTJVAPGM can be a lengthy processes.

There is a penalty for using the JIT Compiler instead of Direct Execution. Startup times are greater than Direct Execution, by about 20%, exaggerated on smaller machines. In large environments, where a very large number of classes are loaded or on small resource constrained systems the cost of the extra startup time may exceed the benefits that the JIT will yield. In these cases Direct Execution may be the best for these scenarios.

The JIT also makes it easier 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.

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 DE'd Java program object, and will therefore run the application using the JIT by default. While it is usually possible to configure the server so that it will run with Direct Execution, running with the JIT will generally offer better performance in these environments.

In V5R2 JIT technology runs about 15% faster than Direct Execution at Optimization Level 40. Also most of the improvements made to the JIT in V5R2 are also now available in V5R1, via a PTF.

7.4 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 occasionally 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 runtime properties or by minimizing 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 JIT compiler on .class files*
  In V5R2, it is no longer recommended that Java class files should be converted into direct execution (machine instruction) Java program objects through the CRTJVAPGM command. Instead the JIT compiler should be used, unless startup time is a critical issue. The JAVA/RUNJVA command in V5R2 will, by default, use the Just In Time compilation for classes that have no direct execution program.

- *Relative Performance* :
  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 -- JIT (no transformer)	1.00
Optimization level 40	1.15
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 and V5R2 JDK 1.3.1. Similar magnitudes would be observed on other releases (this difference has been pretty stable, and has been confirmed in similar magnitudes for a completely different program).

- *Use JIT on .zip and .jar files*
  In V5R2, It is no longer recommended that CRTJVAPGM at Optimization Level 40 should be used on zip and jar files. Instead the JIT compiler should be used, unless startup time is a critical issue The JAVA/RUNJVA command in V5R2 will, by default, use the Just In Time compilation for jar and zip that have no direct execution program.

- *Delete the existing hidden program.*
  To determine if your class/zip/jar file has a permanent, hidden program object, use the DSPJVAPGM command. Because the JIT is faster than optimization level 40, there is little point in having a zip or jar file with a transformed program . Do DLTJVAPGM to delete the hidden program. This will greatly reduce the overall program size. DLTJVAPGM does not affect the jar or zip file itself; only the hidden program.

- *If using JIT consider the special property os400.jit.mmi.threshold.*
  This property set the threshold for the MMI of the JIT. Setting this to a small value will result is compilation of the classes at startup time and will increase the start up time. Setting this to a high value will result in a much faster startup time and compilation of the classes will occur one the threshold is reached. However, if the value is set to high then an increased warm-up time may occur since it will take additional time for the classes to be optimized by the JIT compiler. The default value of 2000 is usually OK for most scenarios.

- *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.

- *If using Direct Execution 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 most 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.

- *If using Direct Execution 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. It is recommended that the GCHMAX parameter only be set on systems that are memory constrained. The value for this should be set slightly lower that the amount of memory available in the pool the application is running out of. This can prevent excessive paging and thrashing that can occur if the JVM heap grows significantly larger than the amount of memory available in the pool.

- *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.

7.5 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 eServer 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 eServer 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 eServer Workload Estimator and its WebSphere input display. The eServer 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 eServer 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 eServer Workload Estimator, pay attention to the "complexity" estimate. Here is where the eServer 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 eServer 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.

eServer Workload Estimator

Sizings for selecting the most appropriate iSeries to run Java and/or WebSphere workloads may be estimated using the eServer Workload Estimator. The estimator is accessable by anyone at
http://www.ibm.com/servlet/EstimatorServlet

Chapter 8. Cryptography Performance

Cryptography enables secure e-Business transactions over a network, but this requires more than just secret or confidential data. It also requires data integrity, identity authentication and transaction non-repudiation. Together, cryptographic algorithms, shared/symmetric keys and public/private keys provide the mechanisms to support all of these requirements. This chapter focuses on the way that iSeries cryptographic solutions improve the performance of secure e-Business transactions..

There are many factors that affect iSeries performance in a cryptographic environment. This chapter discusses some of the common factors and offers guidance on how to achieve the best possible performance. Much of the information in this chapter was obtained as a result of analysis experience within the Rochester development laboratory. Many of the performance claims are based on supporting performance measurement and 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.

Much of the information in this chapter is based on the NetPerf workload. This workload is described at the end of chapter 5. NetPerf is a primitive application. It does little more than issue sockets API calls. While this application does a good job of driving cryptographic and communications stack cycles, a real user application will have this magnitude of CPU cycles for only a percentage of the total CPU time. Also, the SSL measurements shown are based on a specific set of cypher suites and public key sizes. Other choices will perform differently.

Cryptography Performance Highlights for V5R2:

- Support for the 2058-001 Cryptographic Accelerator was added. This adapter increases by a factor of seven the number of SSL full handshakes that can be handled by a single iSeries cryptographic adapter.

- The ability to count SSL handshake operations was added to Collection Services. This allows the user to better understand the performance impact of their secure communications traffic.

Cryptography Performance Highlights for V5R1:

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

8.1 iSeries Cryptographic Solutions

Software is available within OS/400 to support a variety of cryptographic services including those required for SSL (Secure Sockets Layer) and VPN/IPSec (Virtual Private Network).

Two hardware based cryptographic offload solutions are available for the iSeries. One is the **IBM 2058 Cryptographic Accelerator (FC 4805)** and the other is the **IBM 4758 Cryptographic Coprocessor (FC 4801).** Both of these offload portions of cryptographic processing from the host CPU. The host CPU issues requests to the accelerator/coprocessor hardware. The hardware then executes the cryptographic function and returns the results to the host CPU. Because these hardware based solutions handle selected compute-intensive functions, the host CPU is available to support other system activity. OS/400 SSL network communications can use these options to dramatically offload cryptographic processing related to establishing an SSL session.

The 4758 Cryptographic Coprocessor can be used in two ways. First, as described above, OS/400 SSL network communications can use the 4758 Coprocessor to offload cryptographic processing related to establishing an SSL session (i.e., authentication handshake). It will support up to 130 RSA private-key handshakes per second. This card also supports triple DES data encryption and MD5 or SHA-1 data integrity checking. In addition, this card provides secure key storage in an on-card tamper resistant hardware security module (HSM). Finally, custom applications can be written to the CCA (Common Cryptographic Architecture) APIs to access on card cryptographic services, including financial PIN processing. Typically, banking and financial applications use the 4758 Coprocessor in this fashion.

The 2058 Cryptographic Accelerator has been optimized to improve the performance of cryptographic processing related to establishing an SSL session. It supports over 1000 RSA private-key handshakes per second. When activated / varied on in an iSeries system, RSA private-key operations are automatically offloaded to it. Keys are securely stored below the OS/400 MI.

8.2 SSL and VPN

Capacity Planning and Performance Data

Table 8.1 provides some rough capacity planning information for communications when using 100 Mbps Ethernet with SSL and VPN. This is based on measurements gathered from iSeries Model 820 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 8.1. V5R2 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	23.3	13.3	9.5	7.6	7.6
Asym. Connect/Request/Response (ACRR) 8K Bytes	4.5	1.2	1.1	.8	.8
Large Transfer (Stream) 16K Bytes	12.7	1.5	1.4	.9	.9

Notes:
- Capacity metrics are provided for nonsecure and each variation of security policy
- Based on measurements with the NetPerf workload using two iSeries 820 models with V5R2
- 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 characterics are changed, performance may differ significantly.
- CPW is the "Relative System Performance Metric" from Appendix C. 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% * 7.6 transactions/second/CPW = 6384 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 and cryptography scenarios. A real user application will have this type of processing as only a percentage of the overall workload. The IBM eServer Workload Estimator, described in Chapter 22, 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.

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

Table 8.2 below illustrates CPU consumption instead of potential capacity. Essentially, this is a normalized inverse of the CPU capacity data from Table 8.1. It gives another view of the impact of choosing one security policy over another for various NetPerf scenarios.

Table 8.2. V5R2 iSeries SSL and VPN Relative CPU Time					
	Relative CPU Time *(Scaled to the Nonsecure baseline for each transaction type)*				
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	1.0x	1.8x	2.5x	3.1x	3.1x
Asym. Connect/Request/Response (ACRR) 8K Bytes	1.0y	3.9y	4.1y	5.8y	5.9y
Large Transfer (Stream) 16K Bytes	1.0z	8.4z	9.0z	14.8z	14.8z

Notes:
- Based on measurements with the NetPerf workload using two iSeries and 820 models with V5R2
- The table data reflects iSeries as a server (not as 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 8.2, 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.

8.3 Cryptographic Coprocessor Performance

This section provides performance information for iSeries running with the 4758-023 Cryptographic Coprocessor (feature code number 4801). 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 8.3 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 8.4 through 8.7 below show performance in these applications. Notice

that the 4758 Cryptographic Coprocessor does not offload the cryptographic processing associated with VPN.

Cryptographic performance is an important aspect of capacity planning, especially for applications using SSL 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 chapter may be used to assist in capacity planning for this complex environment.

The data presented here 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 4758 Cryptographic Coprocessor.

- NetPerf with the ACRR scenario (see workload description in section 5.5) 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 and V5R2, 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 Information about Securing Applications with SSL in the Networking Security topic under the Networking category of the iSeries Information Center (http://www.ibm.com/eserver/iseries/infocenter).

- 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 8.3 were made between similarly configured iSeries model 840 systems over a dedicated Gigabit Ethernet LAN

Table 8.3 - Cryptographic Capacity Planning			
SSL 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 SSL handshake RSA processing is offloaded to the Cryptographic Coprocessor. MD5 hashing and RC4 encryption/decryption is always done in the host CPU. • Server Authentication only			

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 8.4			
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 8.5			
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 8.6			
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 8.7			
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

8.4 Cryptographic Accelerator Offload Performance

The 2058-001 Cryptographic Accelerator (feature code number 4805) offloads portions of cryptographic processing from the host CPU. The host CPU issues requests to the accelerator. The accelerator then executes the cryptographic function and returns the results to the host CPU. Because the accelerator handles selected compute-intensive functions, the host CPU is available to process other system activity. OS/400 SSL (secure sockets layer) network communications can use the 2058 Cryptographic Accelerator to dramatically offload cryptographic processing related to establishing an SSL session (i.e., handshake). Notice that the 2058 Cryptographic Accelerator does not offload the cryptographic processing associated with VPN.

The web-like measurements in Table 8.4 were made between similarly configured iSeries systems over a dedicated Gigabit Ethernet LAN

Table 8.4 - Cryptographic Accelerator Capacity Planning			
SSL Full Handshake Capacity 2058 Cryptographic Accelerator iSeries V5R2 Model 890			
Number of Cryptographic Coprocessors/Accelerators	Number Jobs	Full Handshake Throughput (Transactions/Second)	Server Capacity Metric (Trans/sec per CPW)
No Coprocessor Offload	Multiple	1577	0.1719
One 2048 Accelerator	Multiple	1097	0.936
Four 2048 Accelerators	Multiple	4382	0.932

Notes:
- In the case of coprocessor offload and multiple jobs, enough jobs were run to fully utilize the Cryptographic Accelerator
- Measurements included RSA (1024 bit key and using CRT), MD5 and RC4 (128 bit key)
- Only SSL handshake RSA processing is offloaded to the Cryptographic Accelerator. 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 890-2487 without a Cryptographic Accelerator. This user might wish to service up to 800 full handshake connections per second using less than 20% of an 8-way partition for network processing. According to Appendix C: "CPW, CIW and MCU Values for iSeries", the CPW rating for an 8-way partition in this particular model is approximately 29300 * (8/24) or 9767. Note the following calculation using the Server Capacity Metric from the table above:

*800 connections/second / (9767CPW * 0.1719) = 47%*

This user cannot meet the 20% utilization requirement with the current system configuration

If the same user installed a Cryptographic Accelerator then:

*800 connections/second / (9767CPW * 0.936) = 9%*

Cryptographic offload dropped the utilization of the eight processor partition from 47% to 9%, allowing the utilization objective to be met.

Note that this level of throughput would requires only one 2058 Cryptographic Accelerator.

As another example, suppose this user expects approximately 500 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 8.4:

500 connections/second / 0.1719 transactions/second per CPW = 2909 CPW

Notice that 2909 CPW is 30% (2909/9767) of the total partition CPW rating. 30% of the eight CPU partition will be consumed servicing the secure LAN connection.

Similarly, if cryptographic offload is used:

500 connections/second / 0.936 transactions/second per CPW = 534 CPW

Only 534 CPW will be required to support the expected 500 connections/second transaction rate. Now only 5% instead of 30% of the partition capacity will be consumed servicing full handshake connections. This freed up two (25%*8) of the partition CPUs to be used for other 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.

Interval number	Interval date time	Elapsed interval seconds	Century digit	Name	Job user	Job number
2	020808153100	60	1	NETPERFSVR	WIEGAND	052770
3	020808153200	60	1	NETPERFSVR	WIEGAND	052770

Full SSL server authentications	Fast SSL server authentications	Full SSL svr and client authentications	Fast SSL svr and client authentications
0	0	0	16,235
0	0	0	15,530

Figure 1 - Collection Services Handshake Counters

The examples above illustrated how CPU savings due to the Cryptographic Accelerator can be estimated if the full handshake rate is known. The best way to get the full handshake rate is to use the handshake counters added to Collection Services in V5R2.

Refer to figure 1, Collection Services Handshake Counters. Four counters were added in V5R2 Collection Services to support SSL / Cryptography. Figure 1 shows example handshake information from Collection Services. It shows that job NETPERFSVR running for 120 seconds executed 16,235+15,530 fast (or cached) handshakes with server and client authentication. This works out to be 265 cached handshakes per second. Because they are cached, the Cryptographic Accelerator will not help reduce CPU utilization. If they had been 265 full handshakes per second, then processor cycles would be freed up for other tasks by adding a Cryptographic Accelerator.

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 8.1 reflects regular SSL handshakes for the Connect/Request/Response scenario without a Cryptographic Accelerator installed.. A full SSL handshake can consume over 20 times more CPU than the cached SSL handshake. The impact of the full handshake on CPU utilization can be minimized by using a

Cryptographic Accelerator. Offloading full handshake processing to one of these adapters may save over 90% of the full handshake host CPU requirements.

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 Accelerators must be considered.

8.5 Cryptography Observations, Tips and Recommendations

- The IBM iSeries Workload Estimator, described in Chapter 22, 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

- Use the V5R2 ability to count SSL handshake operations. This Collection Services capability allows the user to better understand the performance impact of their secure communications traffic. Use this tool to count how many full vs. cached handshake per second are being serviced by the server. For additional information about using this feature, see the description of the QAPMJOBMI database file in the Performance ->Applications for Performance Management topic under the Systems Management category of the iSeries Information Center (http://www.ibm.com/eserver/iseries/infocenter).

- 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.

- 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. Referring to tables 8.1 and 8.2, VPN(ESP with RC4/MD5) and VPN(ESP with DES/MD5) use about the same amount of CPU time but RC4 is much more secure.

- 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 8.1 and 8.2. 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 and later releases support 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.

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

- The Cryptographic Coprocessors and Accelerators are highly scaleable within the iSeries environment. One Cryptographic Coprocessor supports 134 full SSL handshakes per second while eight support 1017. This measured throughput using 8 adapters is within 5% of ideal scaling. Similarly one Cryptographic Accelerator supports 1097 full handshakes per second while four support 4382. This is within 99.9% of ideal scaling.

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

- Up to four Cryptographic Accelerators are supported per system.

- Up to eight Cryptographic Coprocessors are supported per system.

- Applications requiring a FIPS 140 certified, tamper resistant module for storing cryptographic keys should use the IBM 4758-023 Cryptographic Coprocessor. The 2058 Cryptographic Accelerator does not store cryptographic keys in a tamper resistent module.

8.6 Additional Information and Contacts

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

IBM Security and Privacy specialists work with customers to assess, plan, design, implement and manage a security-rich environment for your online applications and transactions. These Security, Privacy,

Wireless Security and PKI services are intended to help customers build trusted electronic relationships with employees, customers and business partners. These general IBM security services are described at: http://www.ibm.com/services/security/index.html . General security news and information 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.ibm.com/security/library/wp_secfund.shtml .
IBM Global Services provides a variety of Security Services for customers and Business Partners. Their services are described at: http://www.ibm.com/services/ .

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

Other performance information can be found at the iSeries Performance Management website at: http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm

IBM 2058 Cryptographic Accelerator (FC 4805) and IBM 4758 Cryptographic CoProcessor (FC 4801): iSeries Infocenter at http://www.ibm.com/eserver/iseries/infocenter ---> then select Networking ---> Networking security ---> Cryptographic Hardware ---> 4758 Cryptographic Coprocessor for iSeries or 2058 Cryptographic Accelerator for iSeries

Collection Services Handshake Counters: iSeries Infocenter at http://www.ibm.com/eserver/iseries/infocenter and then Systems Management ---> Performance ---> Applications for Performance Management ---> QAPMJOBMI database file description

iSeries Performance Management Resource Library at http://www.ibm.com/servers/eserver/iseries/perfmgmt/resource.htm

---

Chapter 9. iSeries File Serving Performance

This chapter will focus on iSeries File Serving Performance.

9.1 iSeries File Serving Performance

When going from **V5R1 to V5R2**, customers should not see a significant change in iSeries 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 at:
http://www-1.ibm.com/servers/eserver/iseries/netserver/ for additional information on AS/400 NetServer.

In V5R2, enhancements were made to NetServer to allow Window 2000 to send write request sizes up to 60K instead of the default size of 4K. This will minimize the traffic on the LAN and provide better throughput with the Integrated File System (IFS).

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**.**

V5R2 Note: Tables and Figures were added to reflect the V5R2 numbers in comparison to V5R1. The Server and Client specifications stated below were used for both V5R2 and V5R1 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

- Window 2000 version 5.0.2195, 260528KB RAM, 16.6GB Disk Drive, Intel(R) PRO/100 VE
  Desktop Contection (Ethernet), Half Duplex

Workload
  200MB File Transfer in both directions (Upload and Download using DOS **copy** command)

Measurement Results:
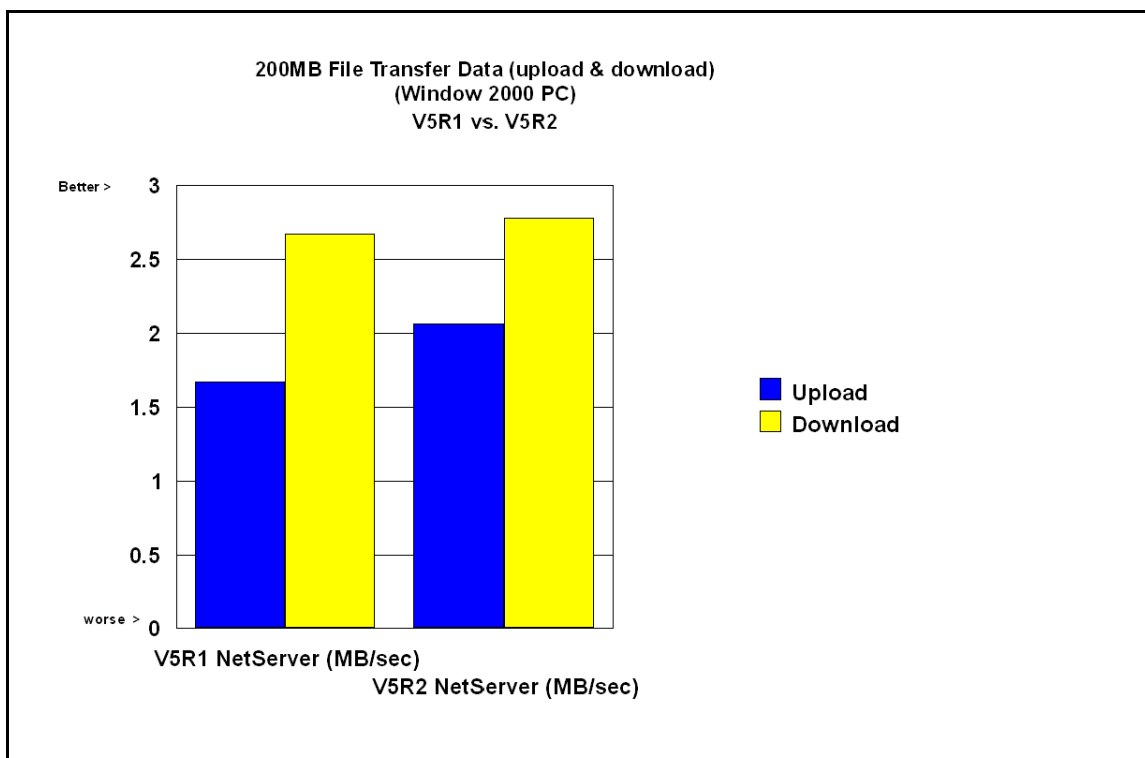
Conclusion/Explanations:

From the chart above in the Measurement Results section, it is evident that when customers goes from
V5R2 to V5R51 (with Window 2000) they can expect to see an improvement in performance. It is also
clear from the chart that independent of what direction the data is flowing one will see an improvement in
performance of large file transfers (download and upload) with iSeries Net Server.

Chapter 10. DB2/400 Client/Server and Remote Access Performance

With V5R2, 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 Appendix C, "CPW, CIW, and MCU Values for iSeries" 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 **V5R2**, NetServer provides file serving performance comparable with that of V5R1.

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.

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. Key factors influencing client performance included processor speed and memory capacity. Client/Server operations that operate on a large amount of data will usually perform better if the client has more than the minimum amount of memory suggested for a particular operating system.

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	314	42	4,418	1,387,252

Query	SQL Statement	Row Size	Columns Per Row	Rows Fetched	Bytes Fetched
	WHERE TRKITEMN <'ITE02210MNUMBR'				

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 iSeries 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 iSeries. While Domino software provides many functions, this section focuses primarily on the performance of mail and calendar server function.

The primary focus of this section will be to discuss the performance characteristics of the iSeries as a server in a Domino environment, and provide capacity planning information and recommendations for optimal performance.

V5R2 Updates (February 2003)
- Domino 6 performance information
- New 800, 810, 825, 870, and 890 models
- Model 810 and 825 iSeries for Domino
- NotesBench audit results

V5R2 Updates (June 2002)
- iNotes Web Access performance information
- New Main Storage Attribute on CHGATTR command and effect on Mail and Calendaring Workload

In general, both the June 2002 and February 2003 versions of OS/400 V5R2 deliver performance equivalent to V5R1 for Domino environments for existing systems.   See Appendix C for information on the performance specifications of the new 890, 870, and 825, 810, and 800 models, including MCU (Mail and Calendar Users) ratings for these models.

This chapter is intended to provide updated performance information for Domino, primarily related to the most recent OS/400 changes. Additional performance information for Domino on iSeries can be found in the following redbooks: (http://www.redbooks.ibm.com/)
- Domino for iSeries Sizing and Performance Tuning,   SG24-5162-01, April 2002
- iNotes Web Access on the IBM eServer iSeries Server,   SG24-6553-00, February 2002
- Upgrading to Domino 6: The Performance Benefits   (At the time of publication of this document, a draft of the redbook is available on-line with the final version expected in January 2003.)

11.1 Workload Descriptions

The Mail and Calendaring Users workload and the iNotes Web Access mail scenarios discussed in this chapter were driven by an automated environment which ran a script similar to the mail workloads from Lotus NotesBench. Lotus NotesBench is a collection of benchmarks, or workloads, for evaluating the performance of Domino servers. The results from the Mail and Calendaring Users and iNotes Web Access workloads are not official NotesBench tests. The numbers discussed for these workloads may not be used officially or publicly to compare to NotesBench results published for other Domino server environments.

Official NotesBench audit results for iSeries are discussed in section *11.13 iSeries NotesBench Audits*. The official iSeries audited NotesBench results can also seen at http://www.notesbench.org .

- **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
  - ❖ Send 1 new mail memos/replies 10Kbytes in size to 3 recipients. (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

- **iNotes Web Access (IWA)**
  Each user completes the following actions an average of every 15 minutes except where noted:

  - ❖ Open mail database which contains documents that are 10 KB in size.
  - ❖ Open the current view
  - ❖ Open 5 documents in the mail file
  - ❖ Send 1 new mail memos/replies 10 KB in size to 3 recipients (every 90 minutes)
  - ❖ Mark one document for deletion
  - ❖ Delete document marked for deletion
  - ❖ Close the view

  The iNotes Web Access workload scenario is similar to the Mail and Calendaring workload except that the Domino mail files are accessed through HTTP from a Web browser and there is no scheduling or calendaring taking place. When accessing mail through Notes, the Notes client performs the majority of the work. When a web browser accesses mail from a Domino server, the Domino server bears the majority of the processing load. The browser's main purpose is to display information.

11.2 Domino R5

Performance information on Domino R5 that was published in previous versions of this document can be accessed at **http://publib.boulder.ibm.com/pubs/html/as400/online/chgfrm.htm .** Much of that information pertained to the early releases of R5. The following R5 information is based on recent performance analysis work with Domino 5.0.9, 5.0.9a, and 5.0.10 and has been tested with V5R1 and V5R2 environments.

1. IOCP Async Mail Notification for Domino 5.0.9, 5.0.9a, and 5.0.10
In Domino version 4, when new mail arrived or when a broadcast message was sent from the console, for all clients with an active session, the new mail or broadcast notification was sent immediately. Every session had its own thread in version 4 which would time out on read waits periodically and check for async sends. When IOCP was introduced in 5.0, Domino went from a thread per connection to a small pool of threads for thousands of connections. With this thread pooling change in version 5, when a thread times out waiting for IO, it isn't associated with any session and the mechanism to do async writes is not active. This function was enabled in Domino 5.0.9 using a polling solution to handle the thread pool

event-driven IOCP model. By default, this poll occurs for all sessions every 10 seconds. The larger the number of sessions, the larger the amount of processing required for the polling. You may want to consider the following options to reduce the processing required for the polling activity:

- Use the following notes.ini variable in the server.ini file to disable this polling mechanism and provide similar function to Domino 5.0.8, IOCP_DISABLE_ASYNC_NOTIFICATION=1 . This makes new mail notification fail and broadcasts fail to sessions which are established but have no active IO on them.

In Domino 5.0.11, improvements were made to the polling algorithm such that the notes.ini variable described above is no longer necessary. This would be true for Domino versions beyond 5.0.11 as well, including Domino 6.

2. Notes_SHARED_DPOOLSIZE environment variable setting
In the past this chapter has recommended leaving the default setting of 12000000 for this variable. Recent experiments using the Mail and Calendaring Users workload have shown that using a smaller size can reduce CPU processing requirements with no impact to response time. Our experiments tested values from 1048576 to 12000000 over a wide range of users and found that the optimal performance was achieved using 1048576. When tested with a web shopper browser based Domino application with a relatively small number of users, setting this variable to 1048576 not have an effect on performance.

11.3 Domino 6

Domino 6 is showing some very impressive performance improvements, both for workloads we've tested in our lab and for customers who have already deployed Domino 6 on iSeries. In this section we'll provide data showing these improvements based on testing done with the Mail and Calendaring User and iNotes Web Access workloads.

Notes client improvements with Domino 6

Using the Mail and Calendaring User workload, we compared performance using Domino 5.0.11 and Domino 6. The table below summarizes our results.

Domino Version	Number of Mail and Calendaring Users	Average CPU Utilization	Average Response Time	Average Disk Utilization
Domino 5.0.11	3,000	39.4%	26ms	7.1%
Domino 6	3,000	27.6%	18ms	5.2%
Domino 5.0.11	8,000	69.7%	67ms	25.2%
Domino 6*	8,000	46.7%	46ms	26.1%

* Additional memory was added for this test

The 3000 user comparison above was done on an iSeries model i270-2253 which has a 2-way 450MHz processor. This system was configured with 8 Gigabytes (GB) of memory and 12 18GB disk drives configured with RAID5. Notice the 30% improvement in CPU utilization with Domino 6, along with a substantial improvement in response time.

The 8000 user comparison was done on a model i810-2469 which has a 2-way 750MHz processor. The system had 24 8.5GB disk drives configured with RAID5. In this test we notice a slightly greater than 30%

improvement in CPU utilization as well as a significant reduction in response time with Domino 6. For this comparison we intentionally created a slightly main storage (memory) constrained environment with 8GB of memory available for the 8000 users. We found that we needed to add 13% more memory, an additional 1GB in this case, when running with Domino 6 in order to achieve the same paging rates, faulting rates, and average disk utilization as the Domino 5.0.11 test. In Domino 6 new memory caching techniques are being used for the Notes client to improve response time and may require additional memory.

Both comparisons shown in the table above were made using single Domino partitions. Similar improvements can be expected for environments using multiple Domino partitions.

iNotes Web Access client improvements with Domino 6

Using the iNotes Web Access workload, we compared performance using Domino 5.0.11 and Domino 6. The table below summarizes our results.

Domino Version	Number of iNotes Web Access users	Average CPU Utilization	Average Response Time	Average Disk Utilization
Domino 5.0.11	2,000	41.5%	96ms	<1%
Domino 6	2,000	24.0%	64ms	<1%
Domino 5.0.11	3,800	19.4%	119ms	<1%
Domino 6	3,800	11.0%	65ms	<1%
Domino 5.0.11	20,000	96.2%	>5sec	<1%
Domino 6	20,000	51.5%	72ms	<1%

Notice that Domino 6 provides at least a 40% CPU improvement in each of the iNotes Web Access comparisons shown above, along with significant response time reductions.   The comparisons shown above were made on systems with abundant main storage and disk resources so that CPU was the only constraining factor. As a result, the disk utilization during all of these tests was less than one percent. The purpose of the tests was to compare iNotes Web Access performance using Domino 5.0.11 and Domino 6.

The 2000 user comparison was done on a model i825-2473 with 6 1.1GHz POWER4 processors, 45GB of memory, and 60 18GB disk drives configured with RAID5, in a single Domino partition. The 3800 user comparison used a single Domino partition on a model i890-0198 with 32 1.3GHz POWER4 processors. This system had 64GB of memory and 89 18GB disk drives configured with RAID5 protection. The 20,000 user comparison used ten Domino partitions, also on an i890-0198 32-way system with 1.3GHz POWER4 processors. This particular system was equipped with 192GB of memory and 360 18GB disk drives running with RAID5 protection.

In addition to the test results shown above, many more measurements were performed to study the performance characteristics of Domino 6. One form of tests conducted are what we call "paging curves". To accomplish the paging curves, a steady state was achieved using the workload. Then, over a course of several hours, we gradually reduced the main storage available to the Domino server(s) and observed the effect on paging rates, faulting rates, and response times. These tests allowed us to build a performance curve of the amount of memory available per user versus the paging rate and response time. Based on a paging curve study of the R5iNotes workload on Domino 6, we determined that, similar to the R5Mail workload, some additional memory was required in order to achieve the same rates as with Domino 5.0.11.

11.4 Response Time and Megahertz relationship

In general, Domino-related processing can be described as compute intensive (See Appendix C - for CIW, Compute Intensive Workload, ratings for the iSeries servers.). That is, faster processors will generally provide lower response times for Domino processing. 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 450MHzz and the 170-2409 has 2 processors rated at 255MHz; the 450MHz processor will provide better response time than 510MHz (2 x 255mhz). The 540MHz, 600MHz, and 750MHz 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.
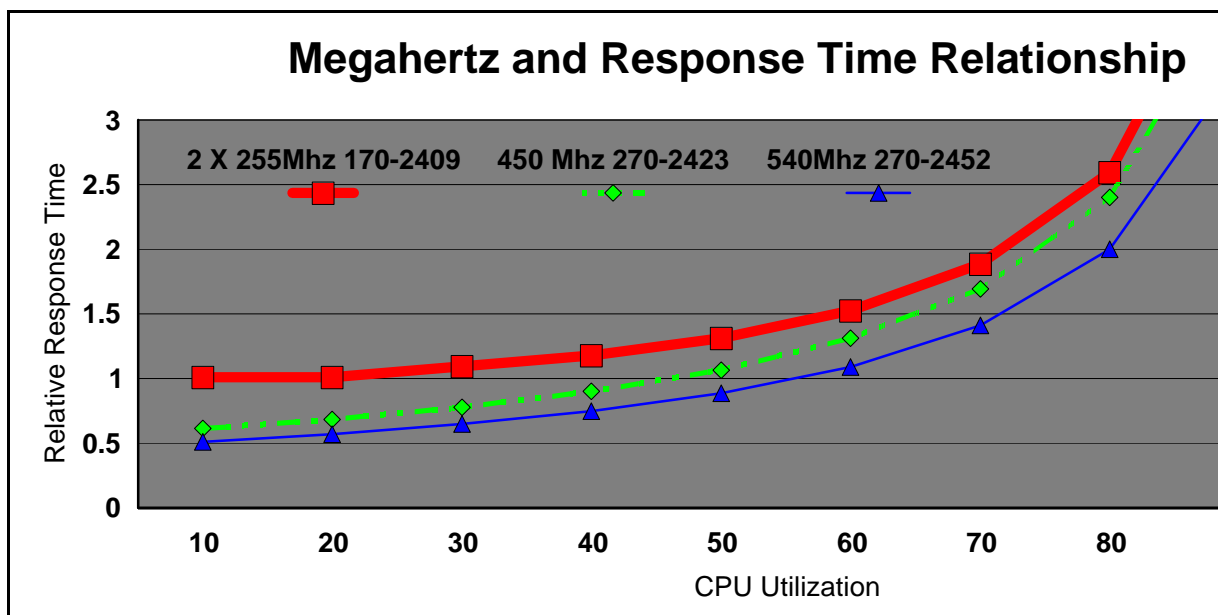
Figure 11.3 Response Time and Megahertz relationship

When using MHz alone to compare performance capabilities between models, it is necessary for those models to have the same processor technology and configuration. Factors such as L2 cache and type of memory controller also influence performance behavior and must be considered. For this reason we

recommend using the tables in Appendix C when comparing performance capabilities between iSeries servers. The data in the Appendix C tables take the many performance and processor factors into account and provides comparisons between the iSeries models using three different metrics, CPW, CIW and MCU.

When comparing models which have different processors types, such as SSTAR and POWER4, it is of particular importance to use the tables in Appendix C or a sizing tool such as the IBM eServer Workload Estimator. The POWER4 processors have been designed to run at significantly higher MHz than SSTAR processors, and the MHz on SSTAR does not compare directly to the MHz on POWER4.

11.5 iSeries for Domino and Dedicated Server for Domino

iSeries for Domino

Included in the V5R2 February 2003 iSeries models are five *iSeries for Domino* offerings. These include three i810 and two i825 models. The iSeries for Domino offerings are specially priced and configured for Domino workloads. With the iSeries for Domino offerings the full amount of DB2 processing is available, and it is no longer necessary to have Domino processing active for non-Domino applications to run well. There are no non-Domino processing guidelines for the iSeries for Domino offerings as with the Dedicated Server for Domino models.

Dedicated Server for Domino

For existing iSeries servers, OS/400 V5R2 (both the June 2002 and the updated February 2003 version) will exhibit similar performance behavior as V5R1 on the Dedicated Server for Domino models. The following discussion of the V5R1 Domino-complimentary behavior applies to V5R2 as well.

The Dedicated Server for Domino (DSD) processor features deliver exceptional price performance for iSeries environments running Domino. The first DSD servers were introduced 8/3/99 and with V5R1, the third generation of DSDs were delivered. Five new V5R1 DSD features provided 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 C, *iSeries CPW Values*.

With V5R1, the DSD was 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 was available after September 28th, 2001, with a 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. 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.ibm.com/servers/eserver/iseries/domino/support** for details on PTFs and QMU levels.

11.6 Performance Tips / Techniques

1.   Be sure to refer to the redbooks listed at the beginning of this chapter which provide Tips and Techniques for tuning and analyzing Domino environments on iSeries.

2.   Our mail tests show approximately a 10% 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. This parameter does not apply to the i825, i870, and i890 servers.

3.   iSeries notes.ini / server document settings:
   *   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. This is in the Server Configuration document for R5.
   *   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. The general rule of thumb is that the larger the buffer pool size, the higher the fault rate, but the lower the cpu cost.   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.

- 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.

4.   OS/400 environment variable settings.
- Notes_SHARED_DPOOLSIZE.   Please refer to the discussion earlier in section 11.2 Domino R5. Based on recent tests we have observed a setting which may provide our environment with improved performance.

- 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.

5.   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.

6.   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.

7.   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.

8.   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.

9.   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.

10.   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.

11.   Full vs. Half duplex on Ethernet LAN.
        Ensure the iSeries 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.

12. Transaction Logging.
Enabling transaction logging typically adds CPU cost and additional I/Os. These CPU and disk costs can be justified if transaction 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 transaction logging active. For the tests with transaction 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.

The redbook listed at the beginning of this chapter, Domino for iSeries Sizing and Performance Tuning, contains an entire chapter on transaction logging and performance impacts.

13. 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.

11.7 iNotes Web Access

The Following are recommendations to optimize your iNotes Web Access environment:

1. Use the default number of 40 HTTP threads. However, if you find that the *Domino.Threads.Active.Peak* is equal to *Domino.Threads.Total*, HTTP requests may be waiting or the HTTP server to make an active thread idle before handling the request. If this is the case for your environment, increase the number of active threads until *Domino.Threads.Active.Peak* is less than *Domino.Threads.Total*. Remember that if the number of threads is set very large, CPU utilization will increase. Therefore, the number of threads should not exceed the peak by very much.

2. Enable *Run Web Agents Concurrently* on the Internet Protocols HTTP tab in the Server Document.

3. For optimal messaging throughput, enable two MAIL.BOX files. Keep in mind that MAIL.BOX files grow as a messages queue and this can potentially impact disk I/O operations. Therefore, we recommend that you monitor MAIL.BOX statistics such as *Mail.Waiting* and *Mail.Maximum.Deliver.Time*. If either or both statistics increase over time, you should increase the number of active MAIL.BOX files and continue to monitor the statistics.

4. Refer to the redbook listed at the beginning of this chapter which contains performance information on iNotes Web Access including the impact of running with SSL.

11.8 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.

11.9 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. After opening the Release Notes for 5.0.3, click on New Enhancements, then Performance Monitoring Statistics.

11.10 V5R2 Main Storage Options

In V5R2 two new attributes were added to the OS/400 CHGATR command, *DISKSTGOPT and *MAINSTGOPT. In this section we will describe our results testing the *MAINSTGOPT using the Mail and Calendar workload. The allowed values for this attribute include the following:

1. *NORMAL
   The main storage will be allocated normally. That is, as much main storage as possible will be allocated and used. This minimizes the number of disk I/O operations since the information is cached in main storage. If the *MAINSTGOPT attribute has not been specified for an object, this value is the default.

2. *MINIMIZE

The main storage will be allocated to minimize the space used by the object. That is, as little main storage as possible will be allocated and used. This minimizes main storage usage while increasing the number of disk I/O operations since less information is cached in main storage.

3. *DYNAMIC

The system will dynamically determine the optimum main storage allocation for the object depending on other system activity and main storage contention. That is, when there is little main storage contention, as much storage as possible will be allocated and used to minimize the number of disk I/O operations. And when there is significant main storage contention, less main storage will be allocated and used to minimize the main storage contention. This option only has an effect when the storage pool's paging option is *CALC. When the storage pool's paging option is *FIXED, the behavior is the same as *NORMAL. When the object is accessed through a file server, this option has no effect. Instead, its behavior is the same as *NORMAL.

These values can be used to affect the performance of your Domino environment. As described above, the default setting is *NORMAL which will work similarly as V5R1. However, there is a new default for the block transfer size of stream files which are created in V5R2. Stream files created in V5R2 will use a block transfer size of 16k bytes, versus 32k bytes in V5R1 and earlier. Files created prior to V5R2 will retain the 32k byte block transfer size. To change stream files created prior to V5R2 to use the 16k block transfer size, you can use the CHGATR command and specify the *NORMAL attribute. Testing showed that the 16k block transfer size is advantageous for Domino mail and calendaring function which typically accesses less than 16k at a time. This may affect the performance of applications that access stream files with a random access patterns. This change will likely improve the performance of applications that read and write data in logical I/O sizes smaller than 16k. Conversely, it may slightly degrade the performance of applications that read and write data with a specified data length greater than 16k.

The *MINIMIZE main storage option is intended to minimize the main storage required when reading and writing stream files and changes the block transfer size of the stream file object to 8k. When reading or writing sequentially, main storage pages for the stream file are recycled to minimize the working set size. To offset some of the adverse effects of the smaller block transfer size and the reduce likelihood that a page is resident, *MINIMIZE synchronously reads several pages from disk when a read or write request would cause multiple page faults. Also, *MINIMIZE avoids reading data from disk when the block of data to be written is page aligned and has a length that is a multiple of the page size.

The *DYNAMIC main storage option is intended to provide a compromise between the *NORMAL and *MINIMIZE settings. This option only has an effect when the storage pool is set to *CALC. The Expert Cache feature of the iSeries allows the file system read and write functions to adjust their internal algorithms based on system tuning recommendations. A system with low paging rates will use an algorithm similar to *NORMAL, but when the paging rates are too high due to main storage contention, the algorithm used will be more like *MINIMIZE. When specifying *DYNAMIC, the block transfer size is set to 12k, midway between the value of *NORMAL and *MINIMIZE.

Deciding when it is appropriate to use the CHGATR command to change the *MAINSTGOPT for a Domino environment is not necessarily straightforward. The rest of this section will discuss test results of using the various attributes. For all of the V5R2 test results show here for the *MINIMIZE and *DYNAMIC attributes, the CHGATR command was used to change all of the user mail .NSF files being used in the test.

The following is an example of how to issue the command:

CHGATR OBJ(name of object) ATR(*MAINSTGOPT) VALUE(*NORMAL, *MINIMIZE, or *DYNAMIC)

The charts below illustrates V5R2 *NORMAL, V5R2 *MINIMIZE, and V5R2 *DYNAMIC paging curve:
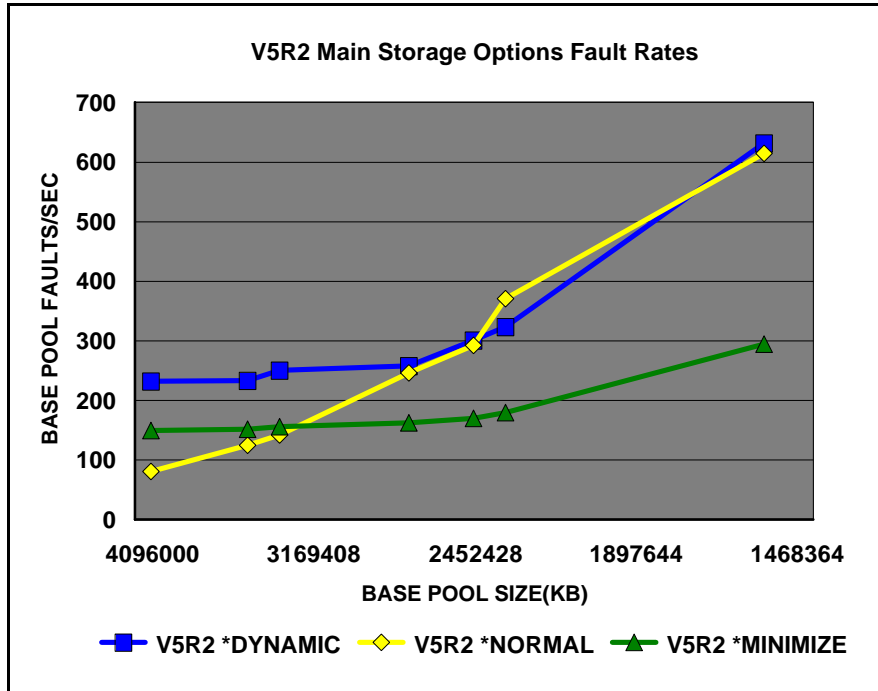
Figure 11.4 V5R2 Main Storage Options - Page Fault Rates

In figures 11.4 through 11.7, results are shown for tests that were performed with a Mail and Calendaring Users workloads running at steady state, and then the amount of main storage available to the *base pool where the Domino server was active was gradually decreased. In Figures 11.4 and 11.5, the tests used an NSF Buffer Pool Size of 300MB. Notice that as the base pool decreases in size, the page faulting will tend to increase. Using the *NORMAL attribute provided the lowest fault rates when memory was most abundant at the left side of the curve. Moving to the right on the chart, it shows that less page faulting takes place with the *MINIMIZE storage option compared to the other two options. Less page faulting will generally provide better performance. We did find that using *DYNAMIC for our Mail and Calendaring Users workload did not have as much affect as it might for some other workloads due to the frequency of opening and closing the files. When running with *DYNAMIC, information about how the file is being accessed is accumulated for the open instance and adjustments are made for that file based on that data. But when the file is closed and reopened the algorithm essentially needs to start over.

Figure 11.5 below shows the response time data rather than fault rates for the same test shown in Figure 11.4 for the V5R2 attributes *NORMAL, *DYNAMIC, and *MINIMIZE.

V5R2 Main Storage Options Response Times
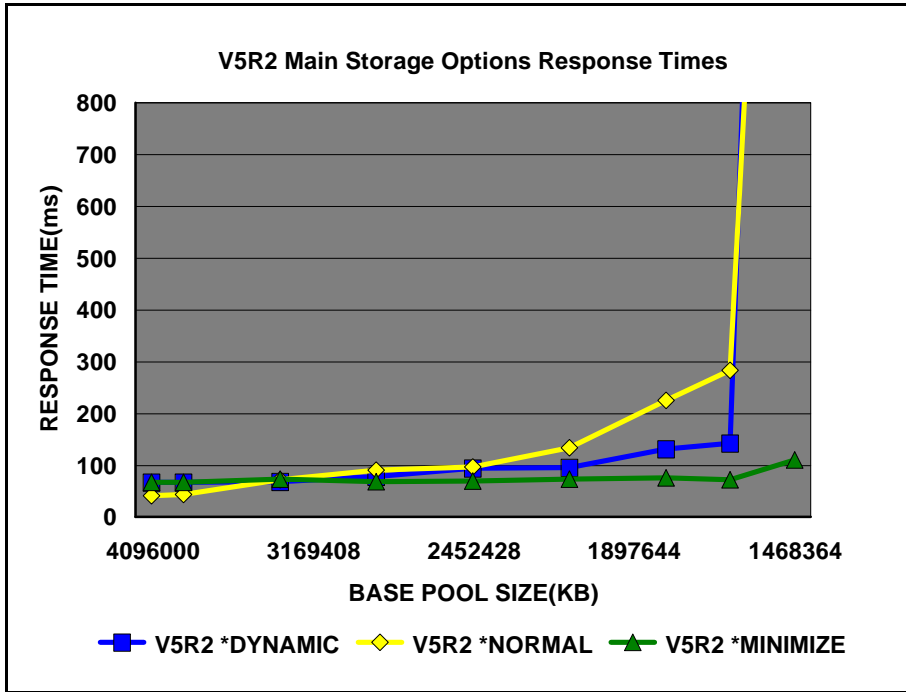
Figure 11.5 V5R2 Main Storage Options - Response Times

Notice that there was not an exact correlation between fault rates and response times as shown in Figures 11.4 and 11.5. The *NORMAL option showed the lowest response times at the lest side of the chart where the most main storage was available. As main storage was constrained, moving to the right on the chart, *MINIMUM provided the lower response times, particularly when main storage became significantly constrained.

The following charts, figures 11.6 and 11.7, compare test data from V5R1 with the V5R2 *NORMAL and *MINIMIZE attributes.

V5R1 and V5R2 Main Storage Options Fault Rates
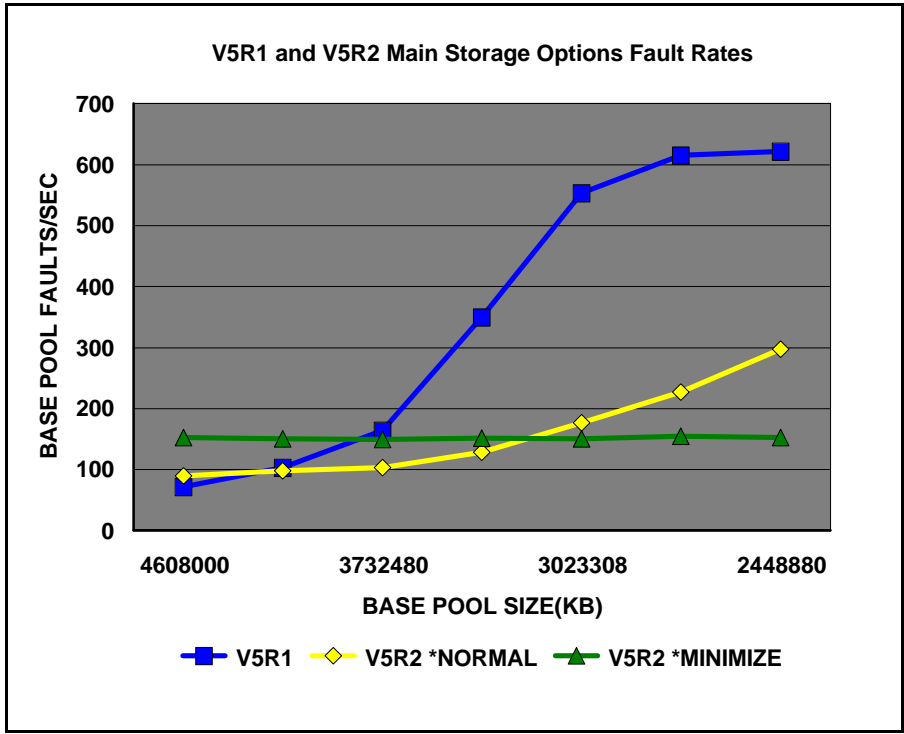
Figure 11.6 V5R1 and V5R2 Main Storage Options - Page Fault Rates

In figures 11.6 and 11.7, the tests used an NSF Buffer Pool size of 750MB. Compared to the data shown in figures 11.4 and 11.5, you will notice that the larger NSF Buffer Pool size caused main storage to become constrained sooner. Figure 11.6 shows that the V5R2 *NORMAL attribute provides a similar faulting rate with V5R1 in the memory rich environment at the left side of the chart. But moving to the right, V5R2 *NORMAL option provided lower faulting rates. *MINIMUM again showed a higher fault rate than *NORMAL in a memory rich pool, but provided lower fault rates when memory was constrained. Note that the faulting rate for V5R1 seems to plateau at the right side of the chart. This is due to the disk drives becoming saturated and not being able to complete and additional faults per second.

Figure 11.7 below shows the response times for the same test described above in figure 11.6.

V5R1 and V5R2 Main Storage Options Response Times
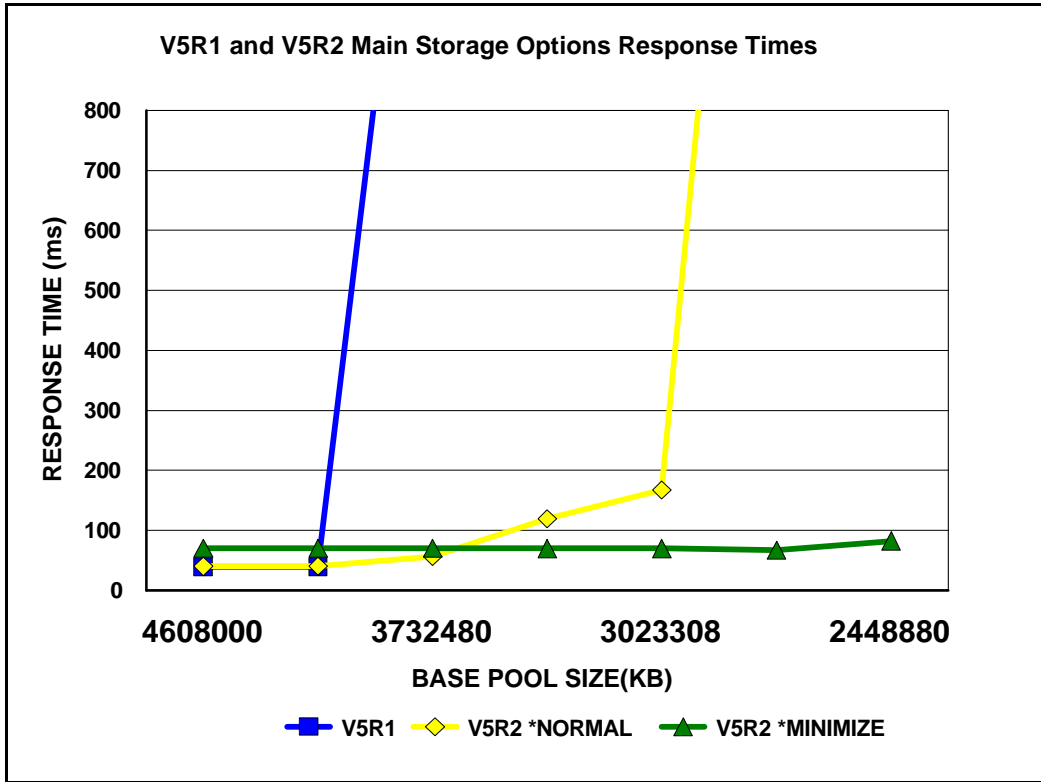
Figure 11.7 V5R1 and V5R2 Main Storage Options - Response Times

The response time data in the chart above shows that V5R2 will provided better response times as the environment moved from being memory rich at the left of the chart to memory constrained on the right. The V5R2 *NORMAL attribute again showed the lowest response time at the left, while using the *MINIMIZE attribute in the memory constrained environment provided consistent response times in a much smaller storage pool.

11.11 Sizing Domino on iSeries

To compare Domino processing capabilities for the various iSeries servers, you should use the CIW and MCU metrics. Generally, these metrics will provide similar comparisons between models since MCU ratings are generated by the Mail and Calendaring User workload, which is best described as a Compute Intensive Workload. CIW is described in detail in Appendix A, and CIW and MCU ratings for the V5R2 iSeries servers can be found in Appendix C.

For sizing Domino mail and application workloads, the recommended method is the IBM eServer Workload Estimator. This tool was previously called the IBM iSeries Workload Estimator. You can access the Workload Estimator from the Domino on iSeries home page (select "sizing Information") or at this URL: http://www.ibm.com/eserver/iseries/support/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 recently. The estimator's rich help text describes the enhancements in more detail. Some of the recent additions include the following: SameTime Application Profiles for Chat, Meeting, and Audio/Video, Domino 6, Transaction Logging, a heavier mail client type, adjustments to take size of database into account, LPAR updates, and enhancement to defining and handling Domino Clustering activity.

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.**

11.12 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 810-2467 is rated at 4200 MCU (at 70% CPU), and the Workload Estimator will indicate that this model supports 2100 users at 100% concurrency. Thus, our 2X factor for MCU versus Typical. Now, if you were to specify 4200 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 810-2467. And by changing the concurrency rating to 100%, the number of users projected at 70% is 2100, or one half of the MCU rating for this server. For the examples in this discussion, we used a mail database size less than 100MB for estimator input.

11.13 iSeries NotesBench Audits

Two NotesBench audits were completed in 2002 using iSeries i890 server configurations.   The results of both audits further established the scaling and performance capabilities of Domino on iSeries.   In August, a NotesBench audit result of 150,000 R5Mail users was published and is the highest number of R5Mail users audited on any platform. This test used 27 Domino partitions with Domino Release 5.0.10. Two of the 27 servers were used as mail routing hub servers and the remaining servers each ran 6,000 R5Mail users. The NotesBench R5Mail workload is comparable to the Mail and Calendaring User workload discussed in this chapter.

In September, a NotesBench audit of 40,200 iNotes Web Access users was published which used the R5iNotes NotesBench workload. This audit also set the record for the highest number of users ever audited for this workload. The R5iNotes workload is comparable to the iNotes Web Access workload discussed in this chapter. This audit also set a record for the lowest response time ever recorded for a NotesBench audit with an average response time of just 58 milliseconds (0.058).   For this audit ten Domino partitions were configured and a beta version of Domino 6 was used. Because this audit was done using a beta release of Domino 6, NotesBench audit rules will cause these results to be withdrawn 6 months from the date of publish.

For both audits, an i890 32-way server was used, but the disk and memory configurations were different. The R5iNotes workload requires substantially more CPU per user than does R5Mail. The following table describes the configurations used for the benchmarks.

NotesBench Workload	Number of Users	i890 32-way config: Memory / Disk Drives	Average CPU Utilization	Average Resposne Time
R5Mail	150,000	265GB / 362x18GB	99%	272ms
R5iNotes	40,200	64GB / 89x18GB	85%	58ms

The full NotesBench audit reports can be accessed at www.notesbench.org . The results can also be viewed on-line at www.ideasinternational.com/benchmark/bench.html#NotesBench .

11.14 Mail and Calendaring Test Data

Performance data using the Mail and Calendaring User workload is provided in section *11.3 Domino 6*. The data in section 11.3 includes comparisons of Domino 5.0.11 and Domino 6 on various processor configurations including models i270 and i810.

The following table provide a summary of measured performance data for existing iSeries models with earlier version of Domino. This chart 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 Users Performance Data							
Mail and Calendar Serving With Domino on iSeries Server Models							
Model	**# Active Notes Users**	**# Domino Partitions**	**Main Storage**	**Response Time (secs)**	**CPU % Busy**	**# Disk Arms**	**Disk % Busy**
840-2461 24w V5R1 ELAN Mirrored (Domino5.06a)	100,500	27 (2 hubs)	128GB	0.07	99.3	270	13.4
820-2458 4w V5R1 ELAN RAID5 (Domino5.06a)	12,000	4	12GB	0.08	71.1	43	17.8
820-2458 4w V5R1 TRLAN RAID5 (Domino5.06a)	6,000	1	12GB	0.03	38.0	20, 2 ASP1,ASP3	7.7 , 0.0 ASP1, ASP3
820-2458 4w V5R1 TRLAN RAID5 Transaction Logging (Domino5.06a)	6,000	1	12GB	0.04	42.3	20 , 2 ASP1,ASP3	11.4 , 8.9 ASP1,ASP3
820-2457 2w V5R1 TRLAN RAID5 (Domino5.06a)	6,750	3	8GB	0.07	71.0	43	6.7
270-2434 2w V5R1 TRLAN RAID5 (Domino5.06a)	6,750	3	8GB	NA	69.6	22	11.3
820-2456 1w V5R1 TRLAN RAID5 (Domino5.06a)	3,250	1	4GB	0.11	73.1	43	1.8

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.

11.15 iNotes Web Access Test Data

Performance data using the iNotes Web Access workload is provided in section *11.3 Domino 6*. The data in section 11.3 includes comparisons of Domino 5.0.11 and Domino 6 on various processor configurations including models i825 and i890.

The following table provide a summary of measured performance data for existing iSeries models with earlier version of Domino. This chart 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. iNotes Web Access Performance Data							
iNotes Web Access Serving With Domino on iSeries Server Models							
Model	# Active iNotes Web Access Users	# Domino Partitions	Main Storage	Response Time (secs)	CPU % Busy	# Disk Arms	Disk % Busy
820-2458 4w V5R1 TRLAN RAID-5 (Domino5.0.8)	2,000	1	3.1 GB	0.133	77	43	1
820-2458 4w V5R1 TRLAN RAID-5 (Domino5.0.8)	2,000	1	1 GB	0.214	79	43	6.3
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

12.1 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.

12.2 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.

12.3 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 Summary

iSeries Linux on LPAR (iSeries Linux) expands the iSeries platform solutions portfolio by allowing customers and software vendors to port existing Linux applications to the iSeries with minimal effort. But, how does it shape up in terms of performance? What does it look like generally and from a performance perspective? How can one best configure an iSeries machine to run Linux?

Key Ideas

- "Linux is Linux." Broadly speaking, Linux on iSeries has the same tools, function, look-and-feel of any other Linux.
- Linux operates in its own independent partition, though it has some dependency on OS/400 for a few key services like IPL ("booting").
- Virtual LAN and Virtual Disk provide differentiation for iSeries Linux.
- Shared Processors (fractional CPUs) provides additional differentiation.
- Linux on iSeries provides a mechanism to port many UNIX and Linux applications to iSeries.
- Linux on iSeries particularly permits Linux-based middleware to exploit OS/400 function and data in a single hardware package.
- Linux on iSeries is available on selected iSeries hardware (see IBM web site for details).
- Linux is not dependent *per se* on OS/400 releases. Technically, any Linux distribution could be hosted by any of the present two releases (V5R1 or V5R2) that allow Linux. It becomes a question of service and support. Users should consult product literature to make sure there is support for their desired combination.
- Linux and other Open Source tools are almost all constructed from a single Open Source compiler known as gcc. Therefore, the quality of its code generation is of significant interest. Java is a significant exception to this, having its own code generation.

13.2 Basic Requirements -- Where Linux Runs

For various technical reasons, Linux may only be deployed on systems with certain hardware facilities.

These are:

- **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, any other Linux partitions. A special software feature called the Hypervisor keeps each partition operating separately.
- **"Guest" Operating System Capability**. This begins in V5R1. 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).

In addition, some models and processor feature codes can run Linux more flexibly than others. The two key features that not all Linux-capable processors support are:

- **Shared Processors.** This variation of LPAR allows the Hypervisor to use a given processor in multiple partitions. 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. This is a large and potentially profitable subject, suitable for its own future paper. Imagine consolidating racks of old, under utilized servers to several partitions, each with a fraction of an iSeries CPU driving it.
- **Hardware Multi-tasking**. This is controlled by the system-wide value QPRCMLTTSK, which, in turn, is controlled by the primary partition. Recent AS/400 and iSeries machines have a feature called hardware multi-tasking. This enables OS/400 (or, now, Linux) to load two jobs (tasks, threads, Linux processes, etc.) into the CPU. The CPU itself will then alternate execution between the two tasks if one task waits on a hardware resource (such as during a cache miss). Due to particular details of some models, Linux cannot run with this enabled. If so, as a practical matter, the entire machine must run with it disabled. In machines where Linux supports this, the choice would be based on experience -- enabling hardware multi-tasking usually boosts throughput, but on occassion would be turned off.

Which models and feature codes support Linux at all and which enable the specific features such as shared processors and hardware multi-tasking are revealed on the IBM iSeries Linux web site.

13.3 iSeries Linux Technical Overview

iSeries Linux Environment
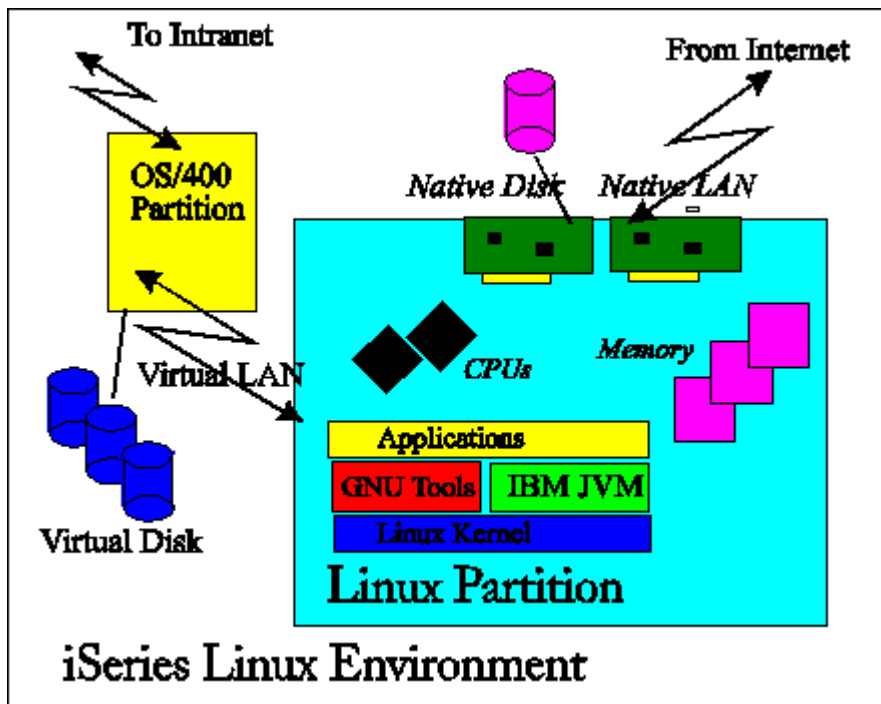
iSeries Linux Architecture

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 the original PowerPC architecture. They do *not* have access to the single level store and OS/400 facilities. To reach OS/400 facilities 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 and, ordinarily, RAIDed.

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 application partition could provide a LAN interface to the outer fire wall. It 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 to obtain OS/400 services like data base. This could be done as three total Linux partitions and an OS/400 partition in the back-end.

See "The Value of Virtual LAN and Virtual Disk" for more on the virtual facilities.

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, PHP, etc.) are also supported. These have their own history and performance implications, but we can do no more than acknowledge that here. There are a couple of generic issues worth highlighting, however.

Applications running in iSeries Linux work in ASCII. At present, no Linux-based code generator supports EBCDIC nor is that likely. 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 in a given protocol. Typically, the translation costs are on the order of five percent of the total CPU costs, usually on the OS/400 side.

iSeries Linux, as a regular Linux distribution, 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. Costs for translating to and from Unicode, if present, will also be around five percent, but this will be comparable to other Linux solutions.

13.4 Basic Configuration and Performance Questions

Since, by definition, iSeries Linux means at least two independent partitions, questions of configuration and performance get surprisingly complicated, at least in the sense that not everything is on one operating system and whose overall performance is not visible to a single set of tools.

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 or with Linux running much I/O and the OS/400 partition extremely CPU-bound.

The first machine will tend to run as expected. If Linux has 3 of 4 CPUs, it will consume about 0.75 of the machine's CPW rating. In many cases, it will more accurately be observed to consume 0.75 of the CIW rating (processor bound may be better predicted by CIW, absent specific history to the contrary).

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 independently operated.
- Virtual disk, generally a good thing, may result in OS/400 and Linux fighting each other for disk access. This is normal if one simply were deploying two traditional applications on an iSeries, but the partitioning may make this more difficult to observe. 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. Tasks representing Licensed Internal Code may show more activity, but attributing this to Linux is not straightforward.
- If the OS/400 partition has a 100 per cent busy CPU for long periods of time, the facilities driving the I/O on the OS/400 side (virtual disk, virtual LAN, shared CD ROM) must fight other OS/400 work for the processor. They will get their share and perhaps a bit more, but this can still slow down I/O response time if the '400 partition is extremely busy over a long period of time.

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 and may add "system" level overhead beyond the new jobs alone. In fact, deploying Virtual Disk in a large, existing ASP will normally optimize performance overall, and would be the first choice. Still, problems may be a bit harder to understand if they occur.
- Existing OS/400 guidelines suggest that disk utilization be kept below 42 per cent for non-load source units. That is, controlling disk utilization for both OS/400 and the aggregate Linux Virtual Disks will also control CPU costs. If this can be managed, sharing an ASP should usually work well.
- 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, usually better, segregating the Linux Virtual Disk (using OS/400 Network Storage objects) into a separate ASP.

13.5 General Performance Information and 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.

Computational Performance -- C-based code

A factor not immediately obvious is that most Linux and Open Source code are constructed with a single compler, the GNC (gcc or g++) compiler.

In Linux, computational performance is usually dominated by how the gcc/g++ compiler stacks up against commercial alternatives such as xlc (OS/400 PASE) and ILE C/C++ (OS/400). *The leading cause of any CPU performance deficit for Linux (compared to Native OS/400 or OS/400 PASE) is the quality of the gcc compiler's code generation.* This is widely known in the Open Source community and is independent of the CPU architecture.

Generally, for integer-based applications (general commercial):

- OS/400 PASE (xlc) gives the fastest integer performance.
- ILE C/C++ is usually next
- Linux (gcc) is last.

Ordinarily, all would be well within a binary order of magnitude of each other. The difference is close enough that ILE C/C++ sometimes is faster than OS/400 PASE. Linux usually lags slightly more, but is usually not significantly slower.

Generally, for applications dominated by floating point, the rankings change somewhat.

- OS/400 PASE almost always gives the fastest performance.
- Linux and ILE C/C++ often trail substantially. In one measurement, Linux took 2.4 times longer than PASE.

ILE C/C++ floating point performance will be closer to Linux than to OS/400 PASE. Note carefully that most commercial applications *do not* feature floating point.

This chart shows some general expectations that have been confirmed in several workloads.

Fraction of ILE Performance

A bar chart titled "Fraction of ILE Performance" with the y-axis labeled "Relative Performance (Bigger Better)" ranging from 0 to 1.2, and the x-axis labeled "Computational Environment" showing three groups: Linux, ILE, and PASE. Each group has two bars: Integer (red) and Floating Point (green).

- Linux: Integer ≈ 0.8, Floating Point ≈ 0.3
- ILE: Integer ≈ 0.9, Floating Point ≈ 0.3
- PASE: Integer ≈ 1.0, Floating Point ≈ 1.0

Legend: Integer (red), Floating Point (green)

One virtue of the i870, i890, and i825 machines is that the hardware floating point unit can make up for some of the code generation deficit due to its superior hardware scheduling capabilities.

Computational Performance -- Java

Generally, Java computational performance will be dictated by the quality of the JVM used. Gcc performance considerations don't apply (occassional exception: Java Native Methods). Performance on the same hardware with other IBM JVMs will be roughly equal, except that newer JVMs will often arrive a bit later on Linux. The IBM JVM is almost always much faster than the typical open source JVM supplied in many distributions.

Web Serving Performance

Work has been done with web serving solutions. Here is some information (primarily useful for sizing, not performance *per se*), which gives some idea of the web serving capacity for static web serving.

Number of 840 Processors in Partition	0.5	1	2	4
# of web server hits per second, Apache 1.3	514	1,024	1,878	3,755
# of web server hits per second, khttpd	860	1,726	3,984	4,961

Here, a model 840 was subdivided into the partition sizes shown and a typical web serving load was used. A "hit" is one web page or one image. The kttpd is a kernel-based daemon available on Linux which serves only static web pages or images. It can be cascaded with ordinary Apache to provide dynamic content as well. The other is a standard Apache 1.3 installation. The 820 or 830 would be a bit less, by about 10 per cent, than the above numbers.

Network Operations

Here's some results using Virtual and 100 megabit ethernet. This pattern was repeated in several workloads using 820 and 840 processors:

TCP/IP Function	100 megabit Ethernet LAN	Virtual LAN
Transmit Data	50-90 megabits per second	200-400 megabits per second
Make Connections	200-3000 connections per second	1100-9500 connections per second

The 825, 870, and 890 should produce slightly higher virtual data rates and nearly the same 100 megabit ethernet rates (since the latter is ultimately limited by hardware). The very high variance in the "make connections" relates, in part, to the fact that several workloads with different complexities were involved.

We also have more limited measurements on Gigabit ethernet showing about 450 megabits per second for some forms of data transmission. For planning purposes, a rough parity between gigabit and Virtual LAN should be assumed.

Gcc and High Optimization (gcc compiler option -O3)

The current gcc compiler is used for a great fraction of Linux applications and the Linux kernel. At this writing, the current gcc version is ordinarily 2.95, but this will change over time. This section applies regardless of the gcc version used. Note also that some things that appear to be different compilers (e.g. g++) are front-ends for other languages (e.g. C++) but use gcc for actual generation of code.

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.

The Gcc Compiler, Version 3

As noted above, many distributions have versions using the 2.95 gcc compiler. However, as this is being written, a new version of the gcc compiler is being released. By the time the reader sees this document, the new compiler may be commonplace. Preliminary runs on beta versions of gcc version 3 suggest that it will partially, but not entirely remove the performance gap between gcc and xlc in particular. The more recent 3.2 shows more variability and less frequent improvement. To the extent it improves, the gap with ILE should close somewhat. Floating point performance is improved, but proportionately. None of the recommendations in terms of Linux versus other platforms change because the improvement is too inconsistent to alter the rankings, though it bears watching in the future as gcc has more room to improve. This is comparing at -O3 as per the prior section's recommendations.

13.6 Value of Virtual LAN and Virtual Disk

Virtual LAN

Virtual LAN is a high speed interconnect mechanism which appears to be an ordinary ethernet LAN as far as Linux is concerned.

There are several benefits from using Virtual LAN:

- *Performance.* It functions approximately on a par with Gigabit ethernet (see previous section, Network Primitives).
- *Cost.* Since it uses built-in processor facilities accessed via the Hypervisor (running on the hosting OS/400 partition), there are no switches, hubs, or wires to deal with. At gigabit speeds, these costs can be significant.
- *Simplification and Consolidation.* It is easy to put multiple Linux partitions on the same Virtual LAN, achieving the same kinds of topologies available in the real world. This makes Virtual LAN ideal for server consolidation scenarios.

The exact performance of Virtual LAN, as is always the case, varies based on items like average IP packet size and so on. However, in typical use, we've observed speeds of 200 to 400 megabits per second on 600 MHz processors. The consumption on the OS/400 side is usually 10 per cent of one CPU or less.

Virtual Disk

Virtual Disk simulates an arbitrarily sized disk. Most distributions make it "look like" a large, single IDE disk, but that is an illusion. In reality, the disks used to implement it are based on OS/400 Network Storage (*NWSSTG object) and will be allocated from all available (SCSI) disks on the Auxiliary Storage Pool (ASP) containing the Network Storage. By design, OS/400 Single Level Store always "stripes" the data, so Linux files of a nontrivial size are accordingly spread over multiple physical disks. Likewise, a typical ASP on OS/400 will have RAID-5 or mirrored protection, providing all the benefits of these functions without any complexity on the Linux side at all.

Thus, the advantages are:

- *Performance*. Parallel access is possible. Since the data is striped, it is possible for the data to be concurrently read from multiple disks.
- *Reduction in Complexity*. Because it looks like one large disk to Linux, but is typically implemented with RAID-5 and striping, the user does not need to deploy complex strategies such as Linux Volume Management and other schemes to achieve RAID-5 and striping. Moreover, obtaining both strategies (which are, in effect, true by default in OS/400) is more complex still in the Linux environment.
- *Cost*. Because the disk is virtual, it can be created to any size desired. For some kinds of Linux partitions, a single modern physical disk is overkill -- providing far more data than required. These requirements only increase if RAID, in particular, is specified. Here, the Network Storage object can be created to any desired size, which helps keep down the cost of the partition. For instance, for some kinds of middleware function, Linux can be deployed anywhere between 200 MB and 1 GB or so, assuming minimal user data. Physical disks are nowadays much larger than this and, often, much larger than the actual need, even when user/application data is added on.
- *Simplification and Consolidation*. The above advantages strongly support consolidation scenarios. By "right sizing" the required disk, multiple Linux partitions can be deployed on a single iSeries, using only the required amount of disk space, not some disk dictated or RAID-5 dictated minimum. Additional virtual disks can be readily added and they can be saved, copied. etc. using OS/400 facilities.

In terms of performance, the next comparison is compelling, but also limited. Virtual Disk can be much faster than single Native disks. In a really large and complex case, a Native Disk strategy would also have multiple disks, possibly managed by the various Linux facilities available for RAID and striping. Such a usage would be more competitive. But we anticipate that, for many uses of Linux, that level of complexity will be avoided. This makes our comparison fair in the sense that we are comparing what real customers will select between and solutions which, for the iSeries customer, have comparable complexity to deploy.

- 1 disk Intel box, 667 MHz CPU: 5 MB/sec for block writes, 3.4 MB/sec for block reads.
- Virtual Disk, OS/400 1 600 MHz CPU: 112 MB/sec for block writes, 97 MB/sec for block reads

As noted, this is not an absolute comparison. Linux has some file system caching facilities that will moderate the difference in many cases. The absolute numbers are less important than the fact that there is an advantage. The point is: To be sure of this level of performance from the Intel side, more work has to be done, including getting the right hardware, BIOS, and Linux tools in place. Similar work would also have to be done using Native Disk on iSeries Linux. Whereas, the default iSeries Virtual Disk implementation has this kind of capability built-in.

13.7 Top Tips for iSeries Linux Performance

Here's a summary of top tips for improving your Linux iSeries LPAR performance:

- **Keep up to date on OS/400 PTFs for your hosting partition**. This is a traditional, but still useful recommendation. So far, some substantial performance improvements have been delivered in fixes for Virtual LAN and Virtual Disk in particular.

- **Investigate keeping up to date with your distribution's kernel.** Since these are not offered by IBM, this document cannot make any claims whatever about the value of upgrading the kernel provided by

your Linux distributor. That said, it may be worth your while to investigate and see if any kernel updates are provided and whether you, yourself can determine if they aid your performance.

- **If possible, compare your Distribution's versions.** This is a topic well beyond this paper in any detail, but in practice fairly simple. A Linux distributor might offer several versions of Linux at any given moment. Usually, you will wish the latest version, as it should be the fastest. But, if you can do so, you may wish to compare with the next previous version. This would be especially important if you have one key piece of open source code largely responsible for the performance of a given partition. There is no way of ensuring that a new distribution is actually faster than the predecessor except to test it out. While, formally, no open source product can ever be withdrawn from the marketplace, actual support (from your distributor or possibly other sources) is always a consideration in making such a call.

- **Evaluate upgrading to gcc 3 when it becomes available.** At this writing, the 3.2 version of gcc and perhaps later versions are being delivered, but some other version may be more relevant by the time you read these words. Check with your Linux distributor about when or if they choose to make it available. With sufficiently strong Linux skills, you might evaluate and perform the upgrade to this level yourself for some key applications if it helps them. Note also that many distributions will distribute only one compiler. The degree of gcc improvement will therefore become a factor in evaluating your upgrade (see the preceding recommendation).

- **Avoid "awkward" LPAR sizes.** If you are running with shared processors, and your sizing recommends one Linux partition to have 0.29 CPUs and the other one 0.65 CPUs, check again.   You might be better off running with 0.30 and 0.70 CPUs. The reason this may be beneficial is that your two partitions would tend to get allocated to one processor most of the time, which should give a little better utilization of the cache. Otherwise, you may get some other partition using the processor sometimes and/or your partitions may more frequently migrate to other processors. Similarly, on a very large machine (e.g. an 890), the overall limit of 32 partitions on the one hand and the larger number of processors on the other begins to make shared processors less interesting as a strategy.

- **Use IBM's JVM, not the default Java typically provided**. IBM's PowerPC Java for Linux is now present on most distributions or it might be obtained in various ways from IBM. For both function and performance, the IBM Java should be superior for virtually all uses. On at least one distribution, deselecting the default Java and selecting IBM's Java made IBM's Java the default. In other cases, you might have to set the PATH and CLASSPATH environment variables to put IBM's Java ahead of the one shipped with most distributions.

- **For Web Serving, investigate khttpd.** There is a kernel extension, khttpd, which can be used to serve "static" web pages and still use Apache for the remaining dynamic functionality. Doing so ordinarily improves performance

- **Keep your Linux partitions to a 4-way or less if possible.** There will be applications that can handle larger CPU counts in Linux, and this is improving as new kernels roll out (up to 8-way is now possible). Still, Linux scaling remains inferior to OS/400 overall. In many cases, Linux will run middleware function which can be readily split up and run in multiple partitions.

- **Make sure you have enough storage in the machine pool to run your Virtual Disk function**. Often, an added 512 MB is ample and it can be less. In addition, make sure you have enough CPU to handle

your requirements as well. These are often very nominal (often, less than a full CPU for fairly large Linux partition, such as a 4-way), but they do need to be covered. Keep some reserve CPU capacity in the OS/400 partition to avoid being "locked out" of the CPU while Linux waits for Virtual Disk and LAN function.

- **Use Virtual LAN for connections between iSeries partitions whether OS/400 or Linux.** If your OS/400 PTFs are up to date, it performs roughly on a par with gigabit ethernet and has zero hardware cost, no switches and wires, etc.

- **Use Virtual Disk for disk function.** Because virtual disk is spread ("striped") amongst all the disks on an OS/400 ASP, virtual disk will ordinarily be faster. Moreover, with available features like mirroring and RAID-5, the data is also protected much better than on a single disk. Certainly, the equivalent function can be built with Linux, but it is much more complex (especially if both RAID-5 and striping is desired). A virtual disk gives the advantages of both RAID-5 and data "striping" and yet it looks like an ordinary, single hard file to Linux.

- **Use Hardware Multithreading if available.** While this will not always work, Hardware multithreading (a global parameter set for all partitions) will ordinarily improve performance by 10 to 25 per cent. Make sure that it profits all important partitions, not just the current one under study, however. Note that some models cannot run with QPRCMLTTSK set to one ("on") and for the models 825, 879, and 890, it is not applicable.

- **Use Shared Processors, especially to support consolidation**. There is a global cost of about 8 per cent (sometimes less) for using the Shared Processors facility. This is a general Hypervisor overhead. While this overhead is not always visible, it should be planned for as it is a normal and expected result. After paying this penalty, however, you can often consolidate several existing Linux servers with low utilization into a single iSeries box with a suitable partition strategy. Moreover, the Virtual LAN and Virtual Disk provide further performance, functional, and cost leverage to support such uses. Remember that some models do not support Shared Processors.

- **Avoid Shared Processors when their benefits are absent**. Especially as larger iSeries boxes are used (larger in terms of CPU count), the benefits of consolidation may often be present without using Shared Processors and its expected overhead penalty. After all, with 16 or more processors, adding or subtracting a processor is now less than 10 per cent of the overall capacity of the box. Similarly, boxes lacking Shared Processor capability may still manage to fit particular consolidation circumstances very well and this should not be overlooked.

- **Watch your "MTU" sizes on LANs.** Normally, they are set up correctly, but it is possible to mis-match the MTU (transmission unit) sizes for OS/400 and Linux whether Virtual or Native LAN. For Virtual LAN, both sides should be 9000. For 100 megabit Native, they should be 1500. These are the values seen in *ifconfig* under Linux. On OS/400, for historical reasons, the correct values are 8996 and 1496 respectively and tend to be called "frame size." If OS/400 says 1496 and Linux says 1500, they are identical. Also, when looking at the OS/400 line description, make sure the "Source Service Access Point" for code AA is also the same as the frame size value. The others aren't critical. While it is certain that the frame sizes on the same device should be identical, it may also be profitable to have all the sizes match. In particular, testing may show that virtual LAN should be changed to 1500/1496 due to end-to-end considerations on critical network paths involving both Native and Virtual LAN (e.g.

from outside the box on Native LAN, through the partition with the Native LAN, and then moving to a
second partion via Virtual LAN then to another).

Chapter 14. DASD Performance

This chapter discusses DASD subsystems available for the iSeries platform.

14.1 Direct Attach (Native)

Performance of various types of DASD with similar configurations are characterized with the use of a batch commercial type workload (small block reads and writes). Performance comparisons are made between different types of DASD, IOP's and IOA's. All of the current DASD measurements are done with RAID protection enabled. For information on previous DASD units and iSeries systems see a previous version of the Performance Capabilities Reference Manual for the release hardware in question.

14.1.1 Hardware Characteristics

14.1.1.1 Devices

DASD	Size (GB)	RPM	Seek Time (ms)		Latency (ms)	Max Drive Interface Speed (MB/s) when mounted in a given tower	
			Read	Write		5074	5094
6717	8	10K	5.3	6.3	3	80	80
6718	18	10K	4.9	5.9	3	80	80
6719	35	10K	4.7	5.3	3	80	160 #1
4326	35	15K	3.6	4.0	2	Not Supported	160 #1
4327	70	15K	3.6	4.0	2	Not Supported	160 #1
#1 - To run at a rate higher than 80 mb/s you also need the new 2757 or 2782 IOA.							

14.1.1.2 Controllers

I/O Processors (IOP)	Disk Controller (IOA)	Write-Cache / up to compressed	Supports Disk compression	Min/Max # of drives in a RAID set
2842 2843 2844	2763	10 MB	No	4/10
	2782	40 MB	No	3/18
	2748 /4748	26 MB	Yes	4/10
	2778 /4778	26 MB / up to 104	Yes	4/10
	2757	235 MB / up to 757	No	3/18
4327 15K disks DO NOT support Disk compression				

14.1.2 V5R1 Direct Attach DASD

This section discusses the direct attach DASD subsystem performance improvements that are new for the V5R1 release. These consist of the following hardware and software offerings :

- 4778/2778 PCI RAID Disk Unit Controller
- 6719 35 GB 10K RPM DASD

Larger capacity drives can appear to be faster than lower capacity drives in the same environment running the same workload. But that perceived improvement can disappear, or even reverse depending upon the workload (primarily where on the disks the data is physically located).

The PCI RAID Disk Unit Controller (#4778/2778) is a DASD IOA that attaches to the PCI bus in the iSeries models 270 and 8xx, in the PCI Expansion Tower (#5075), and in the PCI I/O Towers (#5074/5079). It provides performance improvements up to 10% when compared to the #2748 IOA by utilizing a Fast Write Cache of 26MB (with compression techniques up to 104MB) and SCSI LVD (Low Voltage Differential Signaling) for SCSI Wide-Ultra2 (80MB) support.

14.1.3 V5R2 Direct Attach DASD

This section discusses the direct attach DASD subsystem performance improvements that are new for the V5R2 release. These consist of the following new hardware and software offerings :
- 2757 SCSI PCI RAID Disk Unit Controller (IOA)
- 2782 SCSI PCI RAID Disk Unit Controller (IOA)
- 2844 PCI Node I/O Processor (IOP)
- 4326 35 GB 15K RPM DASD
- 4327 70 GB 15K RPM DASD

14.1.3.1

Batch CPW Workload Performance Comparison
Compare 2778/2757 & 5074 vs 5094
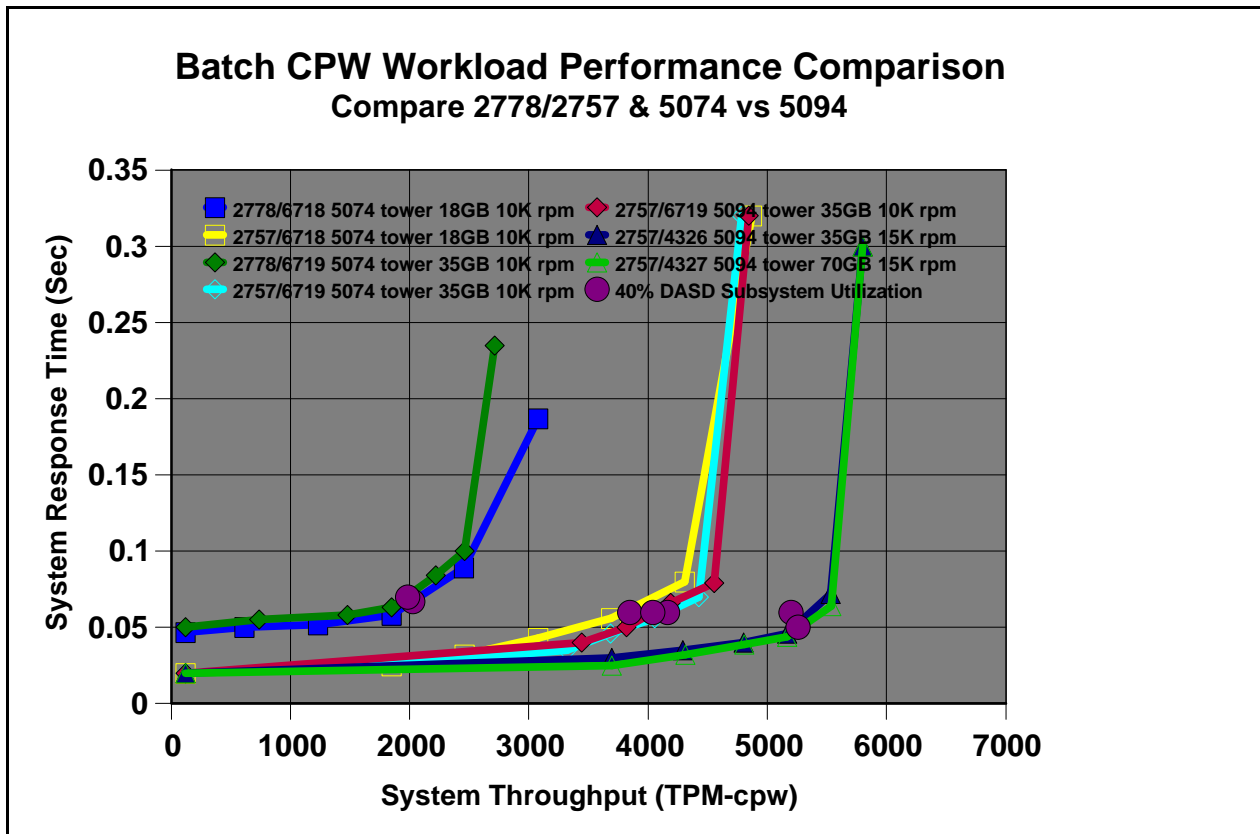
For our workload we attempt to fill the DASD units to between 40 and 50% full so you are comparing units with more actual data, but trying to keep the relative seek distances similar. The reason is that larger capacity drives can appear to be faster than lower capacity drives in the same environment running the same workload in the same size database. That perceived improvement can disappear, or even reverse depending upon the workload (primarily because of where on the disks the data is physically located).

14.1.3.2

Batch CPW Workload
2778 IOA vs 2757 IOA 15 RAID DASD

Legend:
- 2778/6718 10K RPM
- 2778/6719 10K RPM
- 2757/6718 10K RPM
- 2757/6719 10K RPM
- 2757/4326 15K RPM
- 2757/4327 15K RPM

X-axis: System Ops/Sec
Y-axis: DASD Response Time (MS)
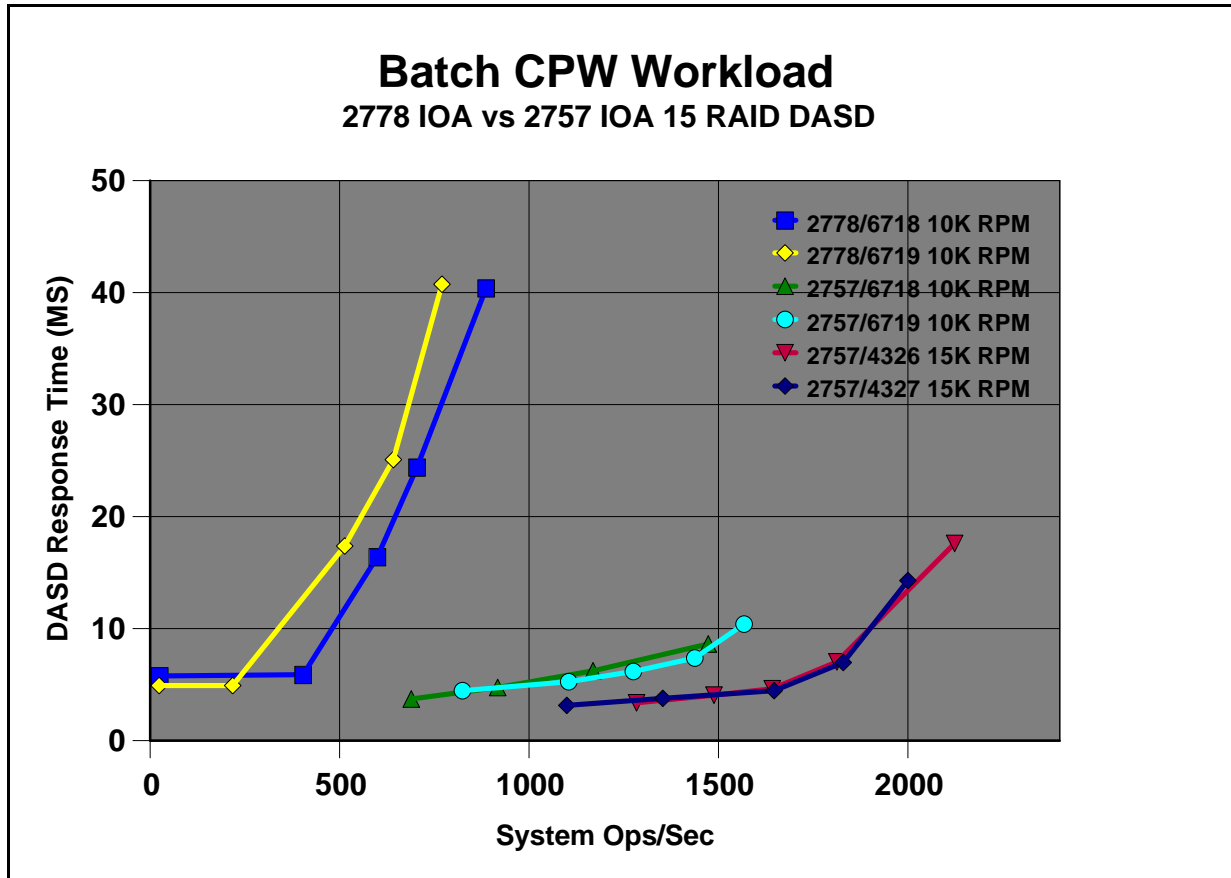
IOA and operation		Number of 35 GB DASD units (Measurement numbers in GB/HR)		
2778 IOA		15 Units	30 Units	45 Units
*SAVF	Save	41	83	122
	Restore	41	83	122
3590E	Save	84	95	110
	Restore	96	95	115
2757 IOA				
*SAVF	Save	82	165	250
	Restore	82	165	250
3590E	Save	95	110	
	Restore	82	115	

This restrictive test is intended to show the effect of the 2757 IOAs in a backup and recovery environment. The save and restore operations to *SAVF (save files) were done on the same set of DASD, meaning we were reading from and writing to the same 15, 30, and 45 DASD units at the same time. So the number of I/O DASD operations are double when saving to *SAVF then when saving or restoring using a tape drive. This was not meant to show what can be expect from a backup environment, but that similar performance was seen in our backup testing environments with ½ the number of DASD units when using the new IOA's.

14.1.4 Direct Attach Observations

One first must understand this is a limited environment in which we are trying to do some simple comparison measurements to give us data so we can give some graphical examples for customers to observe characteristics of new hardware. Our environment is restricted to 1 IOP with 1 IOA and 15 DASD arms under it. We collect performance data using Collection Services and Performance Explorer in order to create our graphs. With our batch commercial type workload (small block reads and writes) we observed the following around the 40% DASD Subsystem Utilization point:

IOP's 2843 vs. 2844:
- No measurable improvements on direct attached DASD. See SAN section for information on IOP improvements.

IOA's 2757 vs. 2778:
- The 2757 achieved up to 2 times better throughput than the 2778 IOA

IOA's 2782 vs. 2763:
- The 2782 IOA achieved up to 25% better throughput at the DASD 40% DASD Subsystem Utilization point than the 2763 IOA measured on 12 DASD units in a 5095 mini tower

5074 tower vs. 5094 tower:
- The 5094 achieved slightly better throughput than the 5074, but the batch CPW workload is not a good indicator in showing the value of the 5094 tower. The new DASD and tower backplanes are capable of large block transfers allowing far greater throughput for large read and write operations. If your workload uses large read and write operations (Such as a Save or Restore operation) you may benefit greatly from the capabilities of the new 5094 tower.

6719 10K RPM DASD units vs. 4326/4327 15K RPM DASD units:
- The 4326/4327 DASD units at the 40% DASD Subsystem Utilization point achieved up to 25% better throughput than the 6719 both on the 2757 IOA

Conclusions:
There can be great benefits in updating to new hardware depending upon the system workload. Most DASD intense workloads should benefit from the new IOAs available. Large block operations will greatly benefit from the new 5094 towers in combination with the new IOA's and DASD units.

14.2 SAN - Storage Area Network (External)

The iSeries is designed for, and optimized around, complex commercial I/O through its distributed I/O processor topology. This architecture uniformly distributes disk I/O and processes the work in parallel. There are many factors to consider when looking at external storage options.

14.2.1 V5R1 Externally attached DASD

• 2766 PCI Fibre Channel Disk Controller Adapter

The PCI Fibre Channel Disk Controller (#2766) is a DASD IOA that attaches to the PCI bus in the 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) 2105/800 DASD via a high-speed Fibre Channel cable length up to 10km. 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.

14.2.2 V5R2 Externally attached DASD

• 2766 2 Gbit link speed FC adapters (The adapter is the same one mentioned in the V5R1 section, the difference being that V5R2 iSeries code enables the adapter to run at 2 Gbit)
• 800 Model ESS
• 2844 IOP

14.2.2.1

Hardware Model	2105 type Disk Models	IOP's Supported	IOA's Supported
E10	A01 (8 GB)	2842	2766
E20	A02 (18 GB)	2843	
F10	A03 (36 GB)	2844	
F20	A04 (70 GB)		
800	A05 (35 GB)		

Comparison of 2844 vs 2843 IOPs
with SAN attached ESS
running a commercial workload

IOP = 2844
IOP = 2843
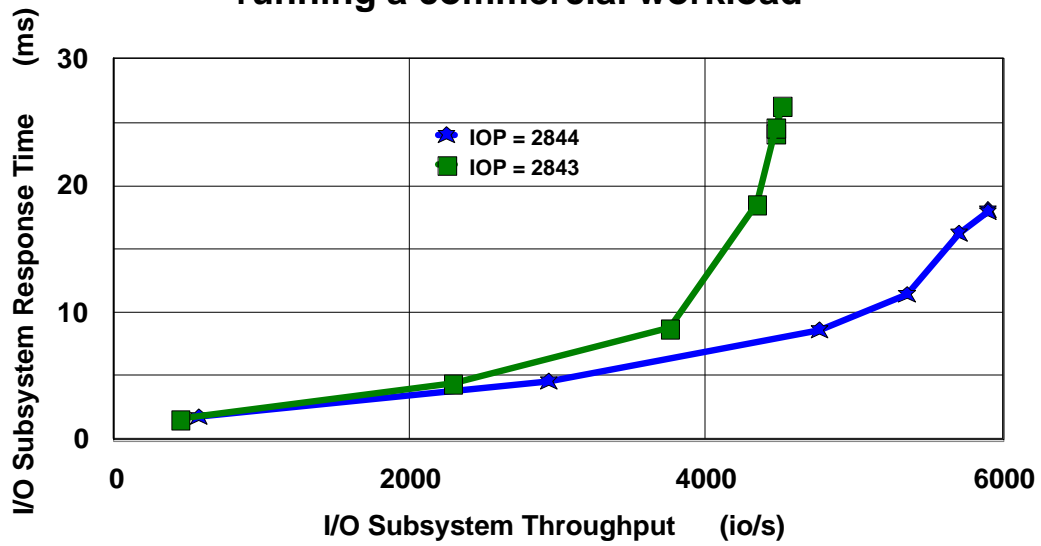
32 ESS F20 LUNs attached via 2 2766 (1Gb) IOAs and 2 IOPs
100% ESS read cache hits

Chapter 15. Save/Restore Performance

This chapter's focus is on the **iSeries models 270 and all 8xx.** For legacy system models, older device attachment cards, and the lower performing backup devices see the V4R5 performance capabilities reference. When the high speed backup devices are attached to the 2765 and 2749 cards, the top rates can dramatically increase from rates on the previous device attachment cards.

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.
- Workload type (Large Database File, User Mix, Source File, integrated file system (Domino, NW STG, 1 Dir M Obj, M Dir M Obj))
- The use of data compression, data compaction, and Optimum Block Size (USEOPTBLK)
- Main Storage Memory and Pool sizes
- Directory structure can have a dramatic effect on save and restore operations.

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 contains backup devices and rates. Later in this document the rates are used to help determine possible performance. A study of some customer data showed that compaction on their database file data occurred at a ratio of approximately 2.7 to 1. The database files used for the performance workloads were created to simulate that result. For integrated file system data we see a general compaction rate of 1.5 to 1.

Table 15.1.1 backup device speed and compaction information		
backup Device	**Rate (MB/S)**	**COMPACT**
DVD-RAM	0.75 Write #3 2.8 Read	2.8 #4
MLR1-S	1.5	1.8
MLR3	2.0	1.8
SLR60	4.0	2.0
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

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.

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.

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 must support it.

15.3 Workloads

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

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

User Mix   **NUMX3GB NUMX6GB NUMX12GB** - The User Mix data is contained in a single library and made up of a combination of source files, database files, programs, command objects, data areas, menus, query definitions, etc. NUMX12GB contains 52900 objects.

Source File   **NSRC1GB** - 96 source files with approximately 30,000 members.

Large Database File   **SR4GB, SR8GB, SR16GB, SR32GB** - The Large Database File workload is a single database file with members 4GB in size to create the database file being tested.

Integrated File System related Workloads:
Analysis of customer systems indicates about 1.5 to 1 compaction on the tape drives with integrated file system data. This is partly due to the fact that the OS/400 programs on the iSeries that store data in the integrated files system, do some disk management functions where they keep the IFS space cleaned up and compressed. And the fact that the objects tend to be smaller by nature, or are mail documents, HTML files or graphic objects that don't compact. The following workloads (1 Dir / M Obj, M Dir / M Obj, Domino, NW STG) show some possible customer integrated file system environments.

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

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

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

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

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

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

But the reality of this formula is that the "LossFromWorkLoadType" is far more complex than described here. The different workloads have different overheads, different compaction rates, and the backup devices use different buffer sizes and different compaction algorithms. The attempt here is to group these workloads as examples of what might happen with a certain type of backup device and a certain workload.

Note: Remember that these formulas and charts are to give you an idea of what you might achieve from a particular backup device. Your data are 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 workloads for concurrent and parallel save and restore operations were performed 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 Database File	95%
User Mix / Domino / NW STG	55%
Source File / 1 Dir M Obj / M Dir M Obj	25%

Example for a DVD-RAM:

DeviceSpeed * LossFromWorkLoad * Compression

$\quad$ 0.75 $\quad$ * $\quad$ 0.95 = (.71) $\qquad$ * 2.8 = (1.995) MB/S * 3600 = 7182 MB/HR = 7.2 GB/HR

$\quad$ 0.75 $\quad$ * $\quad$ 0.95 = (.71) $\qquad$ * No Compression $\quad$ * 3600 = 2556 MB/HR = 2.6 GB/HR

Note: Use 0.95 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 Database File	95%
User Mix / Domino / NW STG	65%
Source File / 1 Dir M Obj / M Dir M Obj	25%

Example for SLR100:

DeviceSpeed * LossFromWorkLoad * Compaction

$\quad$ 5.0 $\quad$ * $\quad$ 0.95 = (4.75) $\qquad$ * 2.0 = (9.5) MB/S * 3600 = 34200 MB/HR = 34.2 GB/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 Database File	95%
User Mix / Domino / NW STG	60%
Source File / 1 Dir M Obj / M Dir M Obj	12%

Example for 3590E:

DeviceSpeed * LossFromWorkLoad * Compaction

$\quad$ 14.0 $\quad$ * $\quad$ 0.95 = (13.3) $\qquad$ * 2.4= (31.92) MB/S *3600 = 114912 MB/HR = 115 GB/HR

$\quad$ 14.0 $\quad$ * $\quad$ 0.60 = (8.4) $\qquad$ * 2.4= (20.16) MB/S *3600 = 72576 MB/HR = 72.6 GB/HR

Example for 3590E Fiber:

DeviceSpeed * LossFromWorkLoad * Compaction

$\quad$ 14.0 $\quad$ * $\quad$ 0.95 = (13.3) $\qquad$ * 2.8= (37.24) MB/S *3600 = 134064 MB/HR = 134.1 GB/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 Database 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**. (**Note: Integrated File System** doesn't support parallel at this time). Understand that the function was designed to help those customers, with very large database files which are dominating the backup window. The goal is to provide them with options to help reduce that window. Large objects, using multiple backup devices, using the parallel function, can greatly reduce the time needed for the object operation to complete as compared to a serial operation on the same object.

Concurrent operations to multiple backup devices will probably be the preferred solution for most customers. The customers will have to weigh the benefits of using parallel verses concurrent operations for multiple backup devices in their environment. The following are some thoughts on possible solutions to save and restore situations:

 - For save and restore with a User Mix and small to medium database 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 large quantity of data and a few very large database 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 database files to multiple backup devices.)

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

 - For systems with a few very large database 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 User 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 database file had 2 members, the 256 GB database file had 4 members, etc. The concurrent testing used duplicate libraries with a single member 32 GB database file in each. The rates obtained are extrapolated from the sample size we use and the time it takes to save that library projecting that to an hour.

Table 15.9.1 V4R5 3590E Parallel Measurements (GB/HR) (Save = S, & Restore = R)											
# Devices		1	2	3	4	5	6	7	8	9	10
128 GB	S	110	215	334	436	534	604				
DB file	R	110	195	290	361	425	448				
256 GB	S						652	743	827	923	
DB file	R						436	548	590	617	
512 GB	S									987	1073
DB file	R									627	661
1 TB	S										987
DB file	R										472

In the table above you find that when the 512 GB database file is projected over an hour we project more than 1 TB per hour but when we use a 1 TB database file it winds up less than 1 TB per hour. This is because when the parallel run 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 quiescent 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 the save the backup devices will fill asynchronously, so the switching shouldn't impact the save time. The second affect of the media switching is that the media may not be evenly filled when the save job is complete. This is an important consideration 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 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).

Table 15.9.2 V4R5 3590E 24 backup Device Measurements (GB/HR)	
Concurrent operations	24 backup devices
Save	2700
Restore	700
Parallel Operation	
Save	2220
Restore	525

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 you are saving a Large Database File workload then the system might be able to achieve the maximum save limit. For an entry model system the limiting factor would be the number of DASD arms which would tend to limit the system to one high speed backup device, or about 100 GB/HR for Large Database File data. A single processor could feed two backup devices with Large Database File data but the number of DASD wouldn't support the operations. If memory and DASD were at their maximum on the mid range and high end systems, the limiting factor would be the processors. For Large Database File data, two backup devices per processor could be driven to capacity.

15.11 How the Number of Processors Affects Performance

With the Large Database 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 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 system architecture does not require that 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 RAID protected DASD units. 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 (GB/HR)				
	16 GB database file Save	16 GB database file Restore	6GB User Mix Save	User Mix Restore
Memory = # Jobs Pool Size Max. Act 10000.00 10000	54	56	40	40
Memory = 2X # Jobs Pool Size Max. Act 10000.00 5000	84	56	58	40
Memory = 3X # Jobs Pool Size Max. Act 10000.00 3300	94	56	62	40

Note: The system value QPFRADJ can be set to 3 = Automatic adjustment, allowing the system to tune itself so that the customer doesn't have to worry about the memory pools.

15.14 How the number of DASD Units affects Performance

The following charts show the affects the number of DASD units have on the save and restore operations. These can help 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 (GB/HR)				
	16 GB database file Save	16 GB database file Restore	6GB User Mix Save	6GB UserMix Restore
15 DASD Units	98	32	56	23
30 DASD Units	98	40	61	29
45 DASD Units	98	59	63	40
60 DASD Units	112	100	75	45
75 DASD Units	112	112	81	45
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 Database 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 database 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 buses and attached DASD towers, there are some conceptual limits that are different than those systems using purely PCI and PCI-X 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 2765 and 2749 cards. The links that attach the migration towers will peak before the links to the PCI or PCI-X 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)

Associate info APAR II12893.

15.17 Saves and Restores Using Save Files

The following rates were achieved on a model 840 24-way iSeries system with 1080 DASD units and 128 GB mainstore. The rates a customer will achieve will depend upon the system resources available. This test was run in a very favorable environment to try to achieve the maximum rates. Software compression rates were gathered using the QSRSAVO API. Data was collected on the same 840 24-way for comparison to a non-compressed save to save file. The CPU used in all compression schemes was near 100%. The compression algorithm cannot span CPUs so the fact that measurements were performed on a 24-way system doesn't affect the software compression scenario.

Table 15.17.1 - Measurements on an 840 24-way system 1080 RAID protected DASD Units (GB/HR)		NSRC1GB	NUMX12GB	SR16GB	Software Compression Ratio
V5R1	Save	19	135	170	
	Restore	7	45	170	
V5R2	Save	19	200	480	
	Restore	7	50	480	
V5R2 Using API DTACPR *LOW	Save		88	108	1.5:1
	Restore		37	57	
V5R2 Using API DTACPR *MED	Save		26	27	2.7:1
	Restore		23	31	
V5R2 Using API DTACPR *HIGH	Save		6	6	3:1
	Restore		39	65	

15.18 *TYPE1 vs. *TYPE2 Directories

15.18.1 *TYPE1 vs. *TYPE2 Directories & Save/Restore PTF (V5R2 - SI05599 / V5R1 - SI05856) (*TYPE2 Directory PTF available for V5R1 see section 15.21 for web site information) V5R2 Measurements on an 840 24-way system <200 RAID protected DASD Units(GB/HR)				
Workload on V5R2 Using a 3590E Tape Drive		*TYPE1 Directories	*TYPE2 Directories No PTF	*TYPE2 Directories With PTF SI05599
1 DIR / Many OBJ	Save	16	16	19
	Restore	5	5	5
Many DIR/ Many OBJ	Save	2.7	5	9.5
	Restore	1.3	2.8	3.4

Example shows where converting to *TYPE2 directories and installing performance PTFs may help customer backup and recovery times.

15.19 V5R2 Rates Small Systems

Table 15.19.1 - V5R2 Measurements on an 270 1-way system 6 RAID protected DASD Units (GB/HR)										
Workload S = Save R = Restore		DVD RAM DTACPR *NO or *DEV	DVD RAM DTACPR *YES	SLR60	SLR100	3580 SCSI	3580 Fiber	3590E SCSI	3590E Fiber	Save to SAVF
NSRC1GB	S	2.5	6.5	10	7.3					
	R	2.6	3.1	4	2.3					
NUMX3GB	S	2.5	6.5		21					
	R	6.6	9.5		10					
NUMX12GB	S			22	22					
	R			10	10					
SR4GB	S	2.6	6.5		32					
	R	9.4	12		19					
SR8GB	S		6.5		33					
	R		13		19					
SR16GB	S			34	34					
	R			17	19					
1 Dir / M Obj	S									
	R									
M Dir / M Obj	S									
	R									
Domino	S									
	R									
NW STG	S									
	R									

15.20 V5R2 Rates Larger Systems

Table 15.20.1 - V5R2 Measurements on an 840 24-way system <200 RAID protected DASD Units (GB/HR)		DVD RAM DTACPR *NO or *DEV	DVD RAM DTACPR *YES	SLR60	SLR100	3580 SCSI	3580 Fiber	3590E SCSI	3590E Fiber	Save to SAVF
Workload S = Save R = Restore										
NSRC1GB	S	2.5	7	17	17	15	16	14	18	19
	R	5.2	6.4	8	8	8.5	8.5	8.8	9	7
NUMX3GB	S	2.5	7			61	70	71	78	
	R	8.8	15			43	40	46	50	
NUMX12GB	S			34	36	68	78	73	78	200
	R			25	25	43	40	46	50	50
SR4GB	S	2.6	7			90	120	108	130	
	R	9.5	19			92	120	111	125	
SR8GB	S		7			99	131	110	130	
	R		19			100	130	112	125	
SR16GB	S			34	36	104	139	114	136	480
	R			34	36	104	139	115	132	480
1 Dir / M Obj	S		2.7		19	19		19		19
	R		5		5	5		5		5
M Dir / M Obj	S		2.7		9	9		9		9
	R		3.4		3.4	3.4		3.4		3.4
Domino	S		2.7		29	85		61		390
	R		9.5		29	85		75		340
NW STG	S		2.7		36	91		90		390
	R		9.5		36	100		90		390

Note: All integrated file system measurements are done on *TYPE2 Directories with PTF SI05599.

15.21 New and Tips on Performance

What's New

- V5R2 - Changes to enhance save and restore times to save files (*SAVF). See Chapter 15.17.
- V5R2 - SLR60, See chapter 15.19 and 15.20 for example numbers
- V5R2 PTF SI05599 can provide integrated file system save improvements up 2X when installed on systems using *TYPE2 directories and in an environment where many directories are involved. Only modest restore improvements, up to 1.25X were seen with this PTF in the many directory environment.

- V5R2 *TYPE2 directories provide integrated file system save and restore time improvements of up to 2X improvement over *TYPE1 directories in an environment where many directories are involved.

- V5R1 - 2765 Fiber card for attaching the 3590E and 3580 Fiber model tape drives.
- V5R1 - 3590E Fiber Channel tape drive, 3580 Fiber Channel tape Drive, DVD RAM drive.
- V5R1 - 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).
- V5R1 PTF SI05856 can provide integrated file system save improvements up 2X when installed on systems using *TYPE2 directories and in an environment where many directories are involved. Only modest restore improvements, up to 1.25X were seen with this PTF.
- V5R1 *TYPE2 directories provide integrated file system save and restore time improvements similar to that documented for V5R2 (see above).

For more information on *TYPE2 directories, see:
(http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm?info/ifs/rzaaxmstv5r1_type2.htm)

TIPS

1. Backup devices are affected by the media type. For most backup devices the right media and density can greatly affect the capacity and speed of your save or restore operation. **USE THE RIGHT MEDIA FOR YOUR BACKUP DEVICE.** (i.e. Use a 25 GB tape cartridge in a 25 GB drive).

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. Domino Online Performance Tips:
   http://www-1.ibm.com/servers/eserver/iseries/service/brms/domperftune.htm

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 V5R1 IPL times and V5R2 IPL times using different hardware configurations. The data for both a normal and abnormal IPL are 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 section provides a simple description of the IPL tests.

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 system activities at the time the system terminates.
- For an abnormal IPL, the benchmark consists of bringing up a database workload and letting it run until the desired number of jobs are running on the system. Once the workload is stabilized, the system is forced to terminate, forcing a mainstore dump (MSD). The dump is then copied to DASD via the Auto Copy function. The Auto Copy function is enabled through System Service Tools (SST). 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.

Large System Benchmark Information

Hardware Configuration
840-2461(24-way) with 128 GB Mainstore
DASD / 1080 units
100 DASD units in ASP 1 RAID protected, 800 DASD units in ASP 2 RAID protected
3 ASP's defined with the remaining DASD units
Mainstore dump was to ASP 2

890-2487 (24-way) with 128 GB Mainstore
DASD / 1080 units
100 DASD units in ASP 1 RAID protected, 800 DASD units in ASP 2 RAID protected
3 ASP's defined with the remaining DASD units
Mainstore dump was to ASP 2

890-2488 (32-way) with 256 GB Mainstore
DASD / 2040 units
100 DASD units in ASP 1 RAID protected, 1500 DASD units in ASP 2 RAID protected
3 ASP's defined with the remaining DASD units
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 load 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

Small System Benchmark Information

Hardware Configuration
270-2250 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 provided 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	
	V5R1 24 Way 840-2461 128 GB MS 1080 Arms	V5R2 24 Way 840-2461 128 GB MS 1080 Arms	V5R2 24 Way 890-2487 128 GB MS 1080 Arms	V5R2 32 Way 890-2488 256 GB MS 2040 Arms	V5R1 1 Way 270-2250 4 GB 6 Arms	V5R2 1 Way 270-2250 4 GB 6 Arms
Hardware	07:20	06:10	11:11	14:51	04:30	03:20
SLIC	09:12	08:37	08:49	13:38	02:30	02:43
OS/400	04:50	06:32	04:55	06:47	01:45	03:09
Total	21:22	21:19	24:55	35:16	08:45	09:12

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	
	V5R1 24 Way 840-2461 128 GB MS 1080 Arms	V5R2 24 Way 840-2461 128 GB MS 1080 Arms	V5R2 24 Way 890-2487 128 GB MS 1080 Arms	V5R2 32 Way 890-2488 256 GB MS 2040 Arms	V5R1 1 Way 270-2250 4 GB 6 Arms	V5R2 1 Way 270-2250 4 GB 6 Arms
Processor MSD	24:42	20:51	36:42	55:44	05:50	05:30
Hardware IPL	04:32	04:00	09:41	13:51	02:50	02:30
SLIC MSD IPL with Copy	2:41:25	1:20:05	1:09:47	02:04:31	06:30	08:37
Shutdown Hardware re-ipl	05:31	05:34	11:00	14:39	04:10	03:13
SLIC re-ipl	12:08	09:02	09:17	14:04	02:55	03:25
OS/400	20:15	20:18	20:52	23:29	02:30	04:17
Total	03:48:33	02:19:50	02:37:19	04:06:18	24:45	27:32
Note: See section 16.4 for changes to data reported for SLIC MSD IPL on large system configurations.						

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.

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

The duration of this recovery step will depend on the type of recovery performed and on the size of the directories. In most cases, a subset directory recovery (SRC C6004250) will be performed which may typically run from 2 minutes to 30 minutes depending upon the system. In rare cases, a full directory recovery (SRC C6004260) may be performed which typically runs much longer than a subset directory recovery.

The duration of the subset directory recovery is dependent on the size of the directory (which relates to the amount of data stored on the system) and on the amount of in-process changes. With the amount of data stored on our largest configurations with one to two thousand disk units, subset directory recovery (SRC C6004250) took from 14 minutes to 50 minutes depending upon the system.

V5R2 includes a significant improvement in storage management directory recovery for high-end systems with large amounts of attached storage. On one system configuration tested with 1,080 disk units, subset directory recovery went from 2 hours and 16 minutes on V5R1 to only 21 minutes on V5R2.

Note: Prior to V5R2, IPL duration's reported for the "SLIC MSD IPL with Copy" phase on the largest system configurations tested were adjusted to reflect typical, projected duration's for storage management directory recovery. The duration's for "SLIC MSD IPL with Copy" in Table 16.3.2 now reflect actual measurements for the systems tested, including those for V5R1. Duration's may be shorter on systems with less storage used.

DASD Unit's 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 varying 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, Windows 2000, and Windows 2003[1] (.NET) with the iSeries. There are several versions of the Integrated xSeries Server:

- The 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 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. The IXS is supported in xSeries models x440, x360, x350, x250, 255, 235, and Netfinity models 7100 and 7600.
- The 1.6 Ghz PCI IXS (#2892-001).
- The 1 Ghz PCI IXS (#2890-003).
- The 850 Mhz PCI IXS (#2890-002).
- The 700 MHz PCI IXS (#2890-001).
- The 333 MHz PCI based Integrated Netfinity Server fits in iSeries 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.

New Integrated xSeries Performance Enhancements in V5R2

- Virtual Ethernet

  The iSeries Virtual Ethernet LAN introduced in V5R1 to enable high speed communications between OS/400 and Linux partitions within the iSeries server is extended in V5R2 to support IXS and xSeries servers attached with IXAs. With this support Windows servers can now communicate with each other and with OS/400 and Linux partitions over the fast, more secure, and reliable Virtual Ethernet LANs. Any new NWSD[2] created in V5R2 will use a new virtual ethernet point to point connection with OS/400. Any pre-V5R2 NWSD[2], when upgraded, will have the old internal LAN connection converted to the faster and more efficient Virtual Ethernet.

  In addition, other Virtual Ethernet ports may be created to connect Integrated xSeries Servers together without the need for external LAN cards. These connections do use the OS/400 CPU, so care must be taken to account for the possible extra load.

- Clustering Service and Shared Disks

  iSeries Windows integration support has been extended to support the Microsoft Clustering Service. With the clustering support provided in Windows 2000 Advanced Server, two IXS or two xSeries servers attached with IXAs can form a cluster and utilize the sixteen new shared storage spaces that are available with V5R2. In the cluster environment if there is a planned or unplanned outage on one of the

---

[1] IBM plans to support Windows 2003 (.NET) Standard and Enterprise servers, when available from Microsoft, on the IXS and IXA attached servers.

[2] Except for Integrated Netfinity Servers (333 & 200 Mhz) which do not support Virtual Ethernet.

Windows servers, the storage spaces can be switched to the second Windows server and the applications can be automatically restarted to reduce the length of the system outage.

- Write Cache Disable

Prior to V5R2, all IXS disk write operations were cached in OS/400 memory buffers. The V5R2 device driver supports the ability to disable write caching should you so desire.

17.2 Configurations

A current list of the supported hardware and software configurations may be found on the IBM Web page:
http://www.ibm.com/servers/eserver/iseries/windowsintegration/overviewhs.htm

In all cases, a separate monitor, keyboard and mouse must be attached to each IXS or IXA attached xSeries server 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. Microsoft NT Server and Windows 2000 Server have not been modified to run on the Integrated xSeries Server. IBM has provided device drivers for the Integrated xSeries Server to access the iSeries' disk, tape and CD-ROM drives. Integrated xSeries Server operations and systems administration are integrated with the iSeries.

The iSeries Integrated xSeries Server runs Microsoft Windows NT Server Version 4.0, or Windows 2000 Server family Standard and Advanced Editions; the same CD-ROM versions that can be purchased from any Microsoft reseller. An IXA attached xSeries Server may only run Windows 2000 Standard and Advanced Server editions.

The Integrated xSeries Adapter for iSeries (IXA) has received the "Designed for Windows 2000 Server" logo as a SCSI Storage Adapter. The Integrated xSeries Server for iSeries and the 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/hwdq/hcl/search.asp then do a search on IBM for product types Storage/SCSI Controller and System/Server Uniprocessor.

The Integrated xSeries Server and Integrated xSeries Adapter do not support VMware ESX Server.

17.3 Effects of Windows loads on the iSeries

There are two ways the IXS and attached xSeries servers create a direct load on the iSeries CPU and subsystems.

- The IXS servers use iSeries storage spaces for its hard drives. This is accomplished by Windows disk device drivers written for the Integrated xSeries Server, and storage space is allocated out of iSeries single level storage. The Windows disk device drivers cooperate with the iSeries OS/400 operating system to perform the disk operations, so iSeries CPU resource is used during disk access. Thus, IXS and IXA operation effects the disk subsystem and the iSeries CPU, and is primarily a function of the disk I/O rate.
- In V5R2, the high speed private LAN and virtual ethernet connections utilize OS/400 CPU during operation. When a virtual ethernet port is used to communicate between Integrated xSeries Servers, or with OS/400 or Linux partitions, the OS/400 CPU is used during the transfer. The amount of CPU used is primarily a function of the number of transactions and their size. However, when the arrival rate of packets is high enough, consolidation of data will improve the operation efficiency and decrease the CPU cost.

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.

Disk I/O CPU Cost

The following chart shows the OS/400 CPU resources required by the Windows server disk operations:

Operation Type	Storage Space Type	Write Cache	CPWs[2] / 1k ops/sec
Random Writes	Static/Dynamic	Enabled	80-90
		Disabled	120-130
	Shared	Enabled	120-130
		Disabled	120-130
Random Reads	n/a	n/a	100-120
Random File Serving[1]	Static/Dynamic	Enabled	105
		Disabled	117
	Shared	Enabled	117
		Disabled	117

[1] File serving operations are mix of operation sizes, and a mix of 80% read and 20% write operations.
[2] A CPW is the "Relative System Performance Metric" from Appendix C. Note that the I/O CPU capacities may not scale exactly by rated system CPW. The CPWs/1k ops/sec value will actually decrease when the number of processors in a iSeries partition increases.

The CPW results above indicate that Static or Dynamic storage spaces with caching enabled cost less than when write cache is disabled, and also less than Shared disks. You can scale the required CPWs linearly more or less based on the expected Windows disk operations per second. The number of disks arms required to meet these performance levels will vary considerably depending a number of factors, such as the protection method (parity or mirroring), the types of disks or disk IOAs, and the size of storage space being accessed.

The IXS and IXA has a maximum disk operation capacity similar to a high end raid controller which might be used in an unattached xSeries configuration. However, the IXS and IXA disk I/O capacity is independent of the data protection method (raid level). The maximum capacity prior to V5R1 is about 3400 Ops/Sec. In V5R2, IOP software improvements have increased the number to about 5000 ops/sec. However, the IXS cards or IXA attached servers are not intended for applications which require a high sustained disk I/O. While you may physically attach many IXSs or IXA attached servers, a single OS/400 partition will limit the combined disk I/O to between 6000-10000 disk ops/sec, depending on a number of factors.

Disk Operation Latency

The average disk operation latency varies depending on the storage space type (Shared, Static or Dynamic), and whether write caching is enabled. The write cache setting may be varied in the hardware manager. Under the individual disk properties is a "Write Cache Enabled" check box. Prior to V5R2, the Write Cache option could not be selected, but write Caching was still enabled by default. With V5R2, write cache is enabled for all Static or Dynamic disks by default. For clustering fail-over integrity reasons, Shared disks have write cache disabled by default.

To give you an idea of the latency scale, the following chart shows some measured average response times[3]:

Average Write Operation Response Time by Configuration
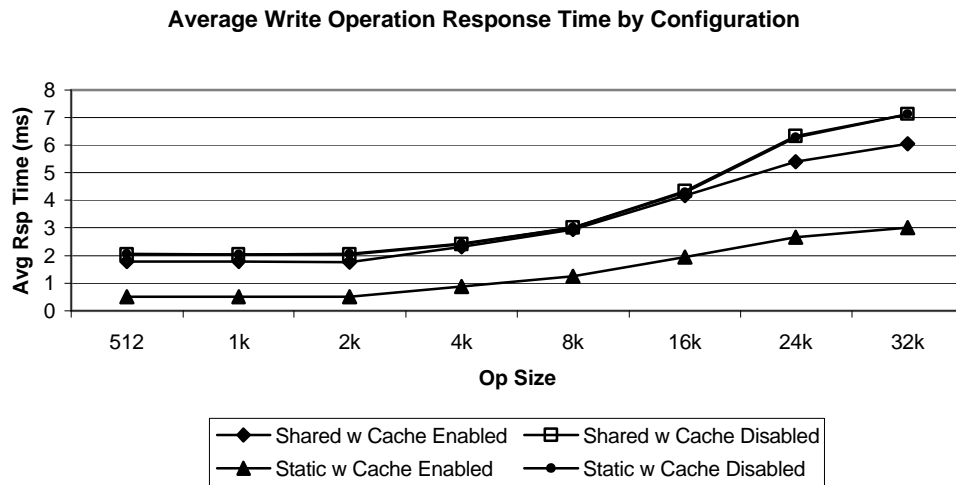
(The performance characteristics of Static and Dynamic disks are identical) As shown above, there may be only a subtle difference in average response time between Shared with cache enabled or disabled, so it is probably not worth enabling the cache for Shared disks. However, the difference will become more pronounced as the storage subsystem becomes busier.

Virtual Ethernet CPU Cost

In V5R2, the private internal LAN connection between an IXS and OS/400 was changed to a higher speed and capacity virtual ethernet connection. The private "point to point" connection is only shared between the Windows server and OS/400. Other virtual ethernet ports (*VRTETH0, etc.) may be assigned which provide a shared virtual LAN between IXS or IXA attached servers and OS/400 partitions.

If the virtual ethernet connections are used for any significant LAN traffic, you may need to account for additional OS/400 CPU requirements. The following table may be used to estimate the potential CPU utilization for virtual ethernet transactions. The chart shows some measured CPW per 1k transactions for streaming operations. For comparison, equivalent results are shown for transfers with external gigabit ethernet cards.

---

[3] Measured on a iSeries 820 Model with V5R2, 10 unprotected 6717 disks, and a 2778 storage controller.

Transfer Source -> Target	Small Transfer (Stream) 16 Bytes CPWs / 1k transactions per sec	Large Transfer (Stream 8k Bytes) CPWs / 1k transactions per sec	Measured Large Transfer Throughput (single connection) Mbps
IXS -> iSeries Point to Point	14.6	61.5	435
iSeries -> IXS Point to Point	27.1	37	365
IXS ext gigabit -> iSeries ext gigabit	9.5	61.2	417
iSeries ext gigabit -> IXS ext gigabit	12.7	58.1	340
IXS virteth -> IXS virteth	18	45.4	453
• Based on the Netperf workload using an iSeries 820 model with V5R2, and IXA attached xSeries Models 255 and 350 • Jumbo frames active on all NICs, including an external gigabit switch. • This is only a rough indicator for capacity planning, actual results may differ.			

The NETPERF TCP_STREAM benchmark is used to drive the workload. Netperf consists of C programs which use a socket connection to read and write data between buffers. The CPW results don't attempt to factor out the minimal application CPU requirements. Thus, the CPU results include the primitive Netperf application, socket, TCP, and ethernet operation costs. A real user application will only have this type of processing as a percentage of the overall workload.

17.4 Summary

The iSeries Integrated xSeries Server with Windows NT Server or Windows 2000 is a full Windows 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 Disks 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.iseries.ibm.com/windowsintegration/

Microsoft Hardware Compatibility Test URL: See http://www.microsoft.com/hwdq/hcl/search.asp
        search on IBM for product types Storage/SCSI Controller and System/Server Uniprocessor.

Redbook: "Direct Attach xSeries for the IBM eServer iSeries Server: A Guide to Implementing xSeries
        Servers in iSeries", SG24-6222 at
        Http://www.redbooks.ibm.com/abstracts/SG246222.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/servers/eserver/iseries/infocenter, then choose "Networking" and
                "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.

V5R2 Additions

In V5R2, some significant items may affect one's LPAR strategy (see "General Tips"):

- "Zero" interactive partitions.   You do not have to allocate a minimum amount of interactive preformance to every partition when V5R2 OS is in the Primary partition.

In V5R2, the customer no longer has to assign a minimum interactive percentage to LPAR partitions (can be 0). For partitions with no assigned interactive capability, LPAR system code will allow interactive as follows: 0.1%   x   (processors in partition/total processors) x processor CPW.

In V5R1, the customer had to allocate a minimum interactive percentage to LPAR partitions as follows: 1.5%   x   (processors in partition/total processors) x   processor CPW. It is expected that the LPAR system code will issue a PTF to change the percentage from 1.5% to 0.5% for V5R1 systems.

Notes:
1. The above formulas yield the minimum ICPW for an LPAR region. The customer still has to divide this value by the total ICPW to get the percentage value to specify for the LPAR partition.
2. If there is not enough interactive CPW available for the partition given the previous formula ... the interactive percentage can be set to the percentage of the (processors in partition/total processors).

General Tips

- Allocate fractional CPUs wisely. If your sizing indicates two partitions need 0.7 and 0.4 CPUs, see if there will be enough remaining capacity in one of the partitions with 0.6 and 0.4 or else 0.7 and 0.3 CPUs allocated. By adding fractional CPUs up to a "whole" processor, fewer physical processors will be used. Design implies that some performance will be gained.

- Avoid shared processors on large partitions if possible. Since there is a penalty for having shared processors (see later discussion), decide if this is really needed. On a 32 way machine, a whole processor is only about 3 per cent of the configuration. On a 24 way, this is about 4 per cent. Though we haven't measured this, the general penalty for invoking shared processors (often, five per cent) means that rounding up to whole processors may actually gain performance on large machines.

V5R1 Additions

In 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.

- Uniprocessor Shared Processors. You can now LPAR a single processor and allocate as little as 0.1 CPUs to a partition. This may be particularly useful for Linux (see Linux chapter).

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, "CPW, CIW and MCU Values for iSeries"*.

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.

LPAR Performance Considerations

Throughput in CPW

- S/Alone 4-W: 1950
- LPAR 2/2 W: 1019 (top), 1025 (bottom) — 5% increase
- 2 x 2-W: 1050 (top), 1050 (bottom) — 3% decrease
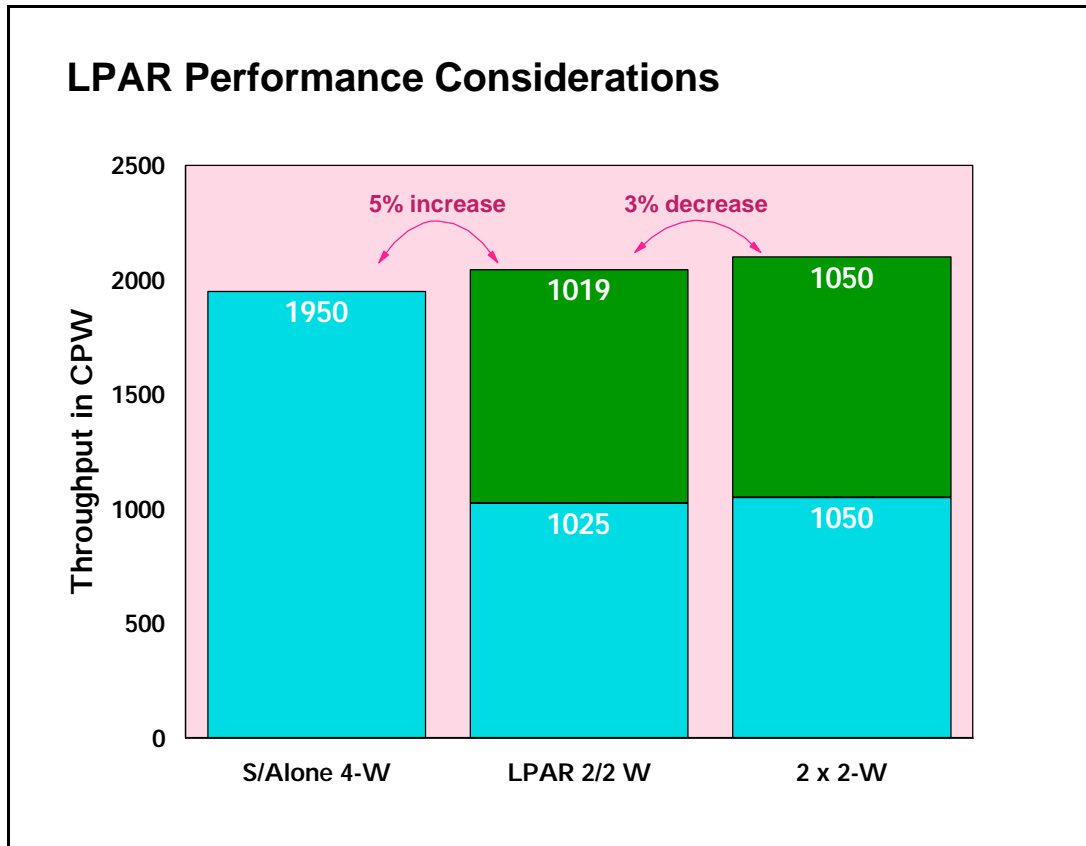
Figure 18.1. LPAR Performance Measured Against Standalone Systems

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.

LPAR Throughput Increase

Total Increase in CPW Capacity of an LPAR System
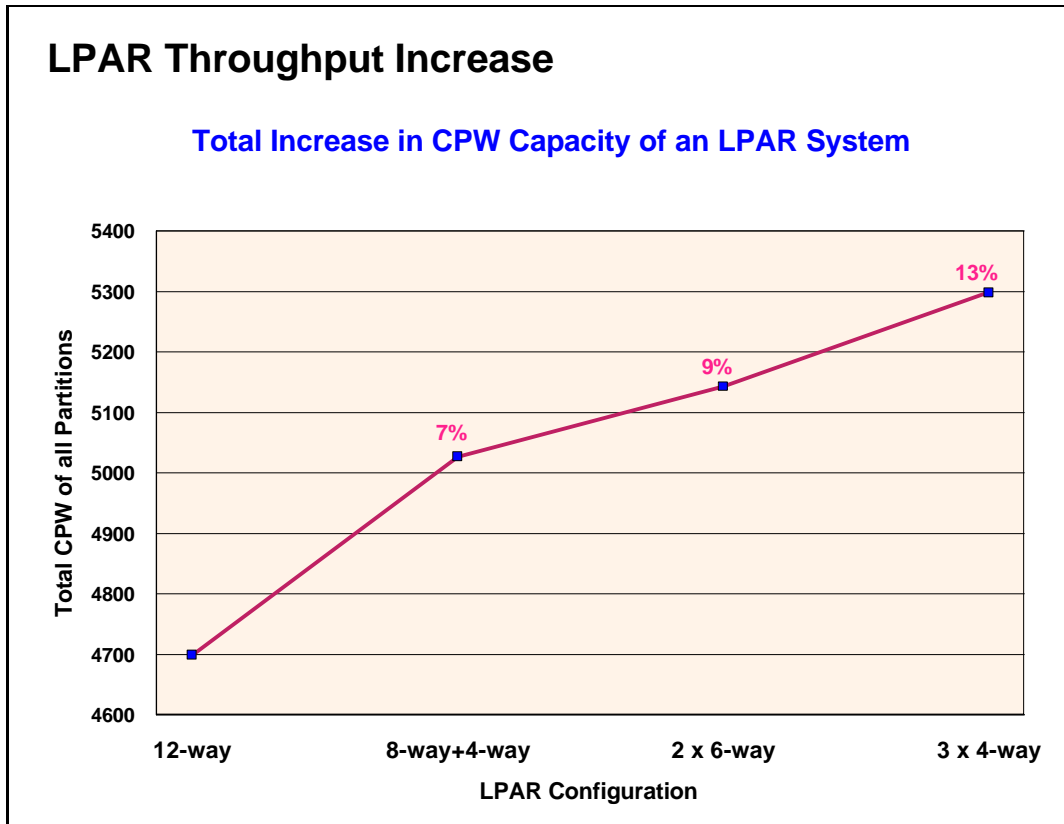
Figure 18.2. 12 way LPAR Throughput Example

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%

	QDYNPTYSCD = '1' (ON)	QDYNPTYSCD = '0'
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.
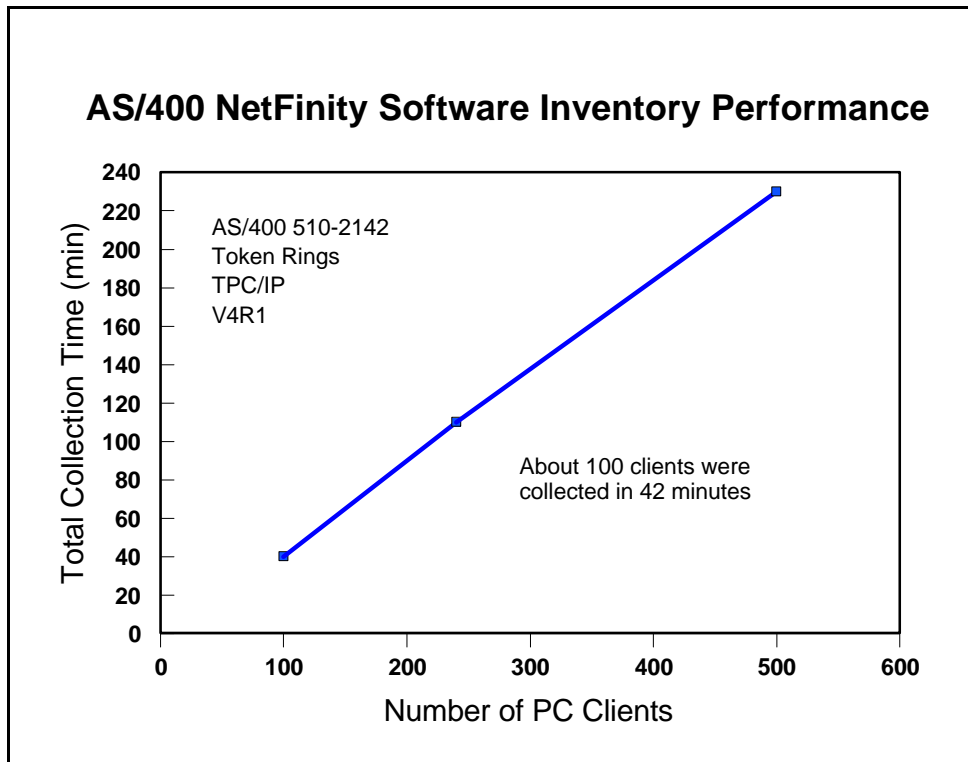
Figure 19.1. AS/400 NetFinity Software Inventory Performance
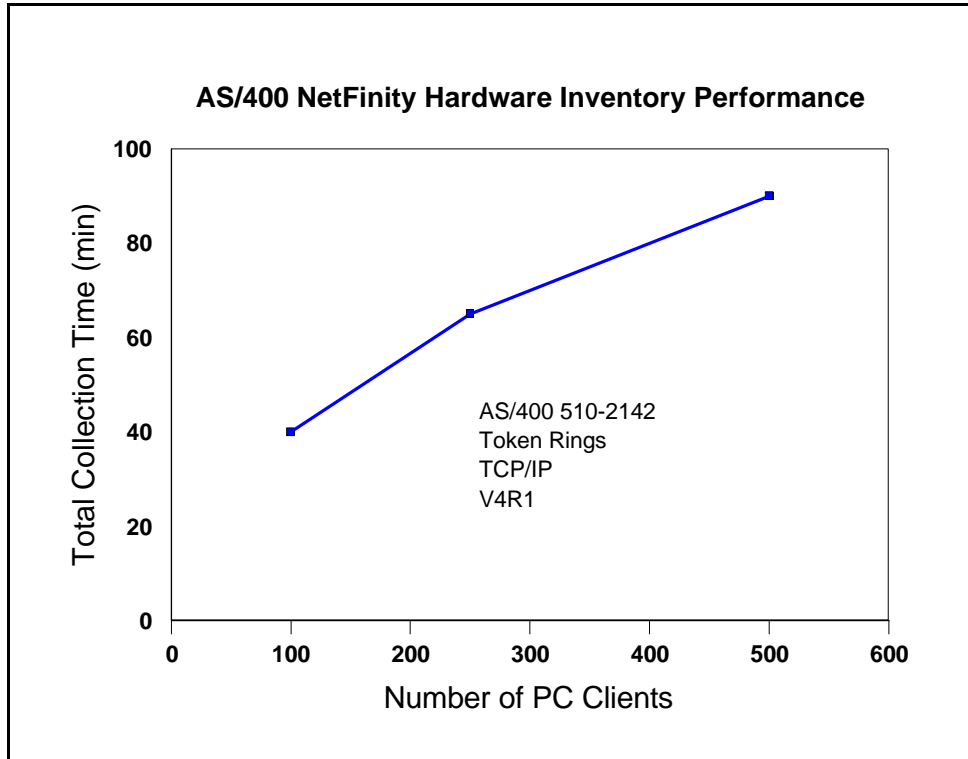
Figure 19.2. AS/400 NetFinity Hardware Inventory Performance

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 iSeries. It may also contain some general guidelines.

20.1 Adjusting Your Performance Tuning for Threads

History

Historically, the iSeries and AS/400 programmers have not had to worry very much about threads. True, they were introduced into the machine some time ago, but the average RPG application does not use them and perhaps never will, even if it is now allowed. Multiple-thread jobs have been fairly rare. That means that those who set up and organize 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 iSeries System Administrators are careless about these distinctions, as it is so easy to do today, poor performance can result as the system moves on to new applications such as Lotus Domino or especially Java.

With Java generally, and with certain applications, it will be commonplace to have multiple threads in a job. That means taking a closer look at some old friends: MAXACT and MAXJOB.

Recall that every subsystem has at least one pool entry. Recall further that, in the subsystem description itself, the pool number is an arbitrary number. What is more important is that the arbitrary number maps to a particular, real storage pool (*BASE, *SHRPOOL1, etc.). When a subsystem is actually started, the actual storage pool (*SHRPOOL1), if someone else isn't already using it, comes to life and obtains its storage.

However, storage pools are about more than storage. They are also about job and thread control. Each pool has an associated value called MAXACT that also comes into play. No matter how many subsystems share the pool, MAXACT limits the total number of threads able to reside and execute in the pool. Note that this is *threads* and not *jobs*.

Each subsystem, also, has a MAXJOBS value associated with it. If you reach that value, you are not supposed to be able to start any more jobs in the subsystem. Note that this is a *jobs* value and not a *threads* value. Further, within the subsystem, there are usually one or more JOBQs in the subsystem. Within each entry you can also control the number of jobs using a parameter. Due to an unfortunate turn in history, this parameter, which might more logically be called MAXJOBS today is called MAXACT. However, it controls *jobs*, not *threads*.

Problem

It is too easy to use the overall pool's value of MAXACT as a surrogate for controlling the number of Jobs. That is, you can forget the distinction between jobs and threads and use MAXACT to control the activity in a storage pool. But, you are not controlling jobs; you are controlling threads.

It is also too easy to have your existing MAXACT set too low if your existing QBATCH subsystem suddenly sees lots of new Java threads from new Java applications.

If you make this mistake (and it is easy to do), you'll see several possible symptoms:

V 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.

V 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.

V 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:

- **V** OPTIMIZE(10) is the lowest and most debuggable.

- **V** OPTIMIZE(20) is a trade-off between rapid compilation and some minimal optimization

- **V** OPTIMIZE(30) provides a higher level of optimization, though it usually avoids the more aggressive options. This level can debug with difficulty.

- **V** 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 CurrencyExchange$_{i,j}$ 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. iSeries PASE Performance

21.1 Introduction

iSeries Portable Application Solutions Environment (iSeries PASE) is designed to expand the iSeries platform solutions portfolio by allowing customers and software vendors to port existing AIX applications to the iSeries with minimal effort.

iSeries PASE is an integrated runtime environment for AIX applications running on the iSeries system. As a native runtime it does not suffer the drawbacks of an emulation environment.

iSeries PASE is designed to accept direct ports from AIX. iSeries 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 iSeries Integrated Language Environment accounted for the majority of C and C++ application ports, many of which originally ran on UNIX. Recently, however, the iSeries 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 iSeries.

iSeries PASE enables the iSeries 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 iSeries PASE.

iSeries PASE is supported on all iSeriese series servers (iSeries systems introduced after 8/97). iSeries PASE takes advantage of the iSeries system and RS/6000's common investment in PowerPC processor technology. The PowerPC processor switches from its normal iSeries mode, in order to execute an application in the iSeries PASE runtime. Applications running in iSeries PASE may need to be enabled to access DB2 Universal Database for iSeries and integrated with iSeries 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 iSeries PASE from PartnerWorld for Developers at http://www.iseries.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).

iSeries PASE Technical Overview

iSeries PASE 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). Programs running in iSeries 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.

iSeries 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 iSeries 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 iSeries PASE program can run any arbitrary sequence of (user-state PowerPC) hardware instructions without jeopardizing system security or integrity.

The iSeries PASE address space provides a mapping from iSeries PASE addresses to addresses in a Teraspace region. Any memory mapped into the iSeries PASE region of Teraspace has exactly the same accessibility (relative address and storage protection) to both iSeries PASE programs and SLS programs. Both SLS segments and unnamed memory can be mapped into Teraspace. iSeries 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). iSeries PASE programs deal with 4-byte (untagged) pointers, in contrast to the 16-byte tagged MI pointers used in SLS.

iSeries PASE currently provides support for 32-bit AIX application, in support of a 32-bit addressing model. The processors used in iSeries systems also support 64-bit applications.

iSeries PASE Run-time Support

V4R5 adds significant support including Xwindows and a large number of shells and utilities.

Applications running in iSeries PASE work in ASCII. The AIX C compiler (xlc) does not support EBCDIC. Any iSeries 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).

iSeries 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 iSeries PASE to handle character encoding conversions for calls from iSeries PASE to arbitrary ILE procedures. iSeries PASE runtime support includes functions iconv_open, iconv, and iconv_close for character encoding conversion.

iSeries 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 iSeries PASE shared library included with OS/400 option 33. A iSeries 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 iSeries PASE runs independently of SLS runtime. The system does not implicitly set any iSeries PASE environment variables, but the Qp2RunPase API allows the caller to specify a set of environment variables to initialize in iSeries PASE, and program QP2SHELL passes a copy of all ILE environment variables to the iSeries PASE program. SLIC implements support for the (AIX) system calls needed to run the C library runtime subset supported by iSeries PASE, and also supports system calls for platform-specific functions such as building a tagged space pointer (_SETSPP) and calling an ILE procedure (_ILECALL).

iSeries PASE Development Environment

iSeries 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 iSeries PASE must be done on an AIX system release that is compatible (and optimally the same as) the AIX release for which the iSeries PASE release provides equivalence. For example, V4R5 iSeries PASE is based on a subset of AIX 4.3.3.

Characteristics of Application Candidates for iSeries PASE

When planning to port an application from AIX to the iSeries there are three choices: you can port to the iSeries Integrated Language Environment (ILE) , Linux on iSeries under LPAR, or you can port to iSeries PASE.

Here are some of the reasons to choose iSeries PASE:
1. If the UNIX/AIX APIs your applications uses are already supported by iSeries PASE, then there is very little application porting to be done. You can determine how well your application conforms to supported iSeries PASE APIs by using the "filtering" tool found at http://www.ibm.com/developer/factory/porting/apitool.html .
2. iSeries PASE provides an environment for running more computationally intensive applications on the iSeries system by providing optimized math libraries.
3. iSeries PASE allows the use of UNIX based build processes, which is especially useful when you have an existing, complicated build process.
4. iSeries PASE supplies support for *fork* and *exec*, which does not currently exist on the iSeries system (except through *spawn* which is significantly different).
5. iSeries PASE is designed to satisfy dependencies on an ASCII character set and to satisfy dependencies on X-Windows support.
6. iSeries 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 iSeries PASE to other environments on iSeries 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 iSeries PASE to AIX and in other instances to compare iSeries PASE to the iSeries Integrated Language Environment (ILE). Ease of porting is an important consideration. When porting from a UNIX environment to iSeries, the software developer is now given two options, either to port to ILE or to use iSeries PASE. The difference between ILE and iSeries PASE is fairly small, assuming that above guidelines are followed. iSeries PASE may be the preferred option, for example, in computationally intensive applications.

CPU Intensive Workloads

While most applications which run on iSeries are commercial by nature, many modern applications have the characteristic of requiring additional CPU capacity. In order to measure the relative performance of iSeries PASE, two workloads were devised that perform numeric intensive calculations. Be aware that these results cannot be used to compare iSeries 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 iSeries 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 iSeries 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 iSeries 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, iSeries 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 iSeries PASE forking compared to AIX forking. The test was run on an iSeries 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 iSeries 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 iSeries was about 5mS (on the above iSeries system). The comparable cost per fork on AIX was about 0.3mS. Although the iSeries 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 iSeries 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 iSeries and a server iSeries. 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 iSeries PASE and ILE.
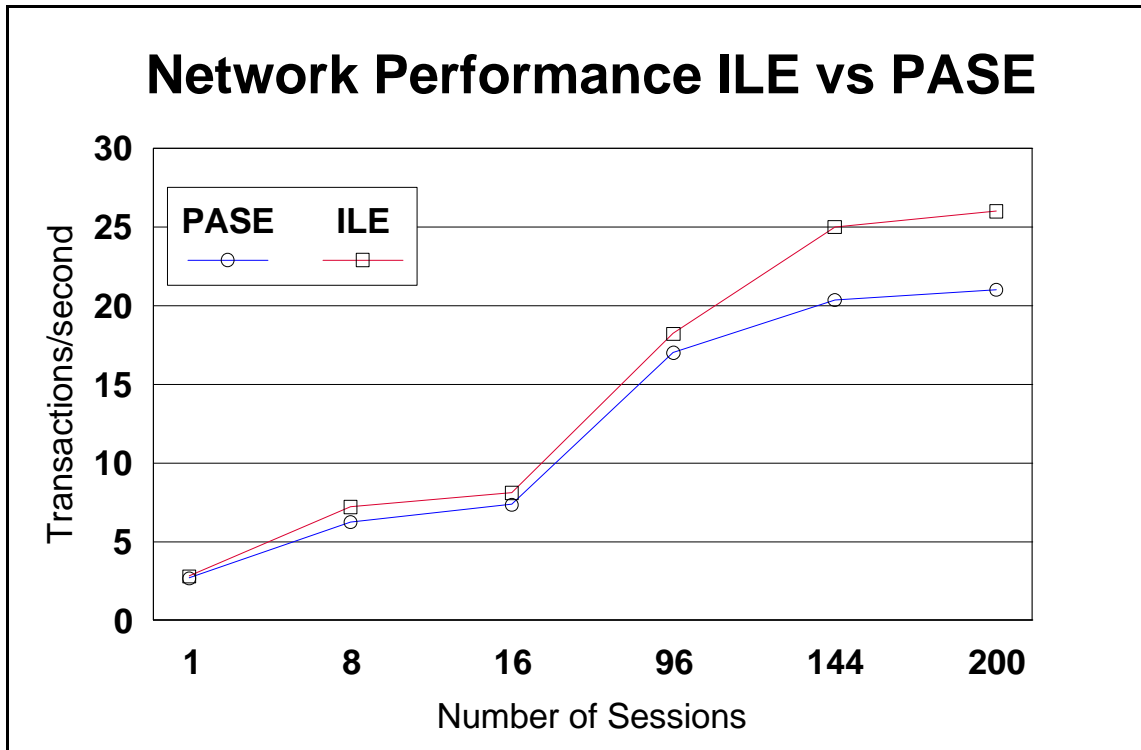
Figure 21.1 Comparison of Network Performance for iSeries PASE and ILE

The performance of iSeries 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 iSeries PASE and ILE being limited by CPU capacity.

Cross Environment Calls

This test was created to determine the overhead of calling ILE functions from iSeries PASE. The iSeries 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 iSeries 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 iSeries 4-way model 730-2067 with #1511 feature) are shown in the Figure 21.2.

**AS/400 PASE to ILE Call Cost
(By Argument Type)**
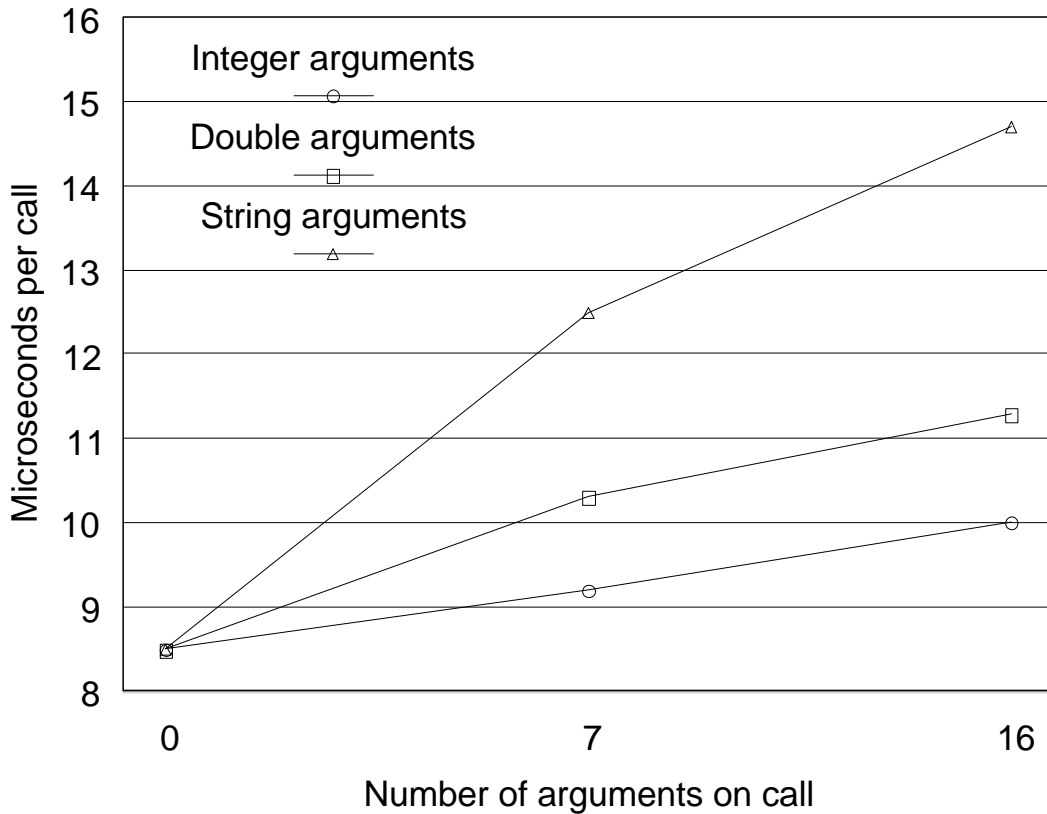
Figure 21.2 Cross environment calls comparing iSeries PASE to ILE and ILE to ILE

As Figure 21.2 shows, the cost of calling ILE from an iSeries PASE program depends on a number of factors including the number and type of arguments. The cost of calling ILE from iSeries PASE is longer than ILE to ILE bound calls. The impact of the cost of calling ILE from iSeries PASE will be a function of how much processing in the iSeries 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 iSeries PASE program to ILE code. If the time in the iSeries 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 iSeries 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

AS/400 PASE PROGRAM

ILE CODE

Time in
AS/400
PASE
program

AS400 PASE TO ILE CALL

Time in
ILE
code
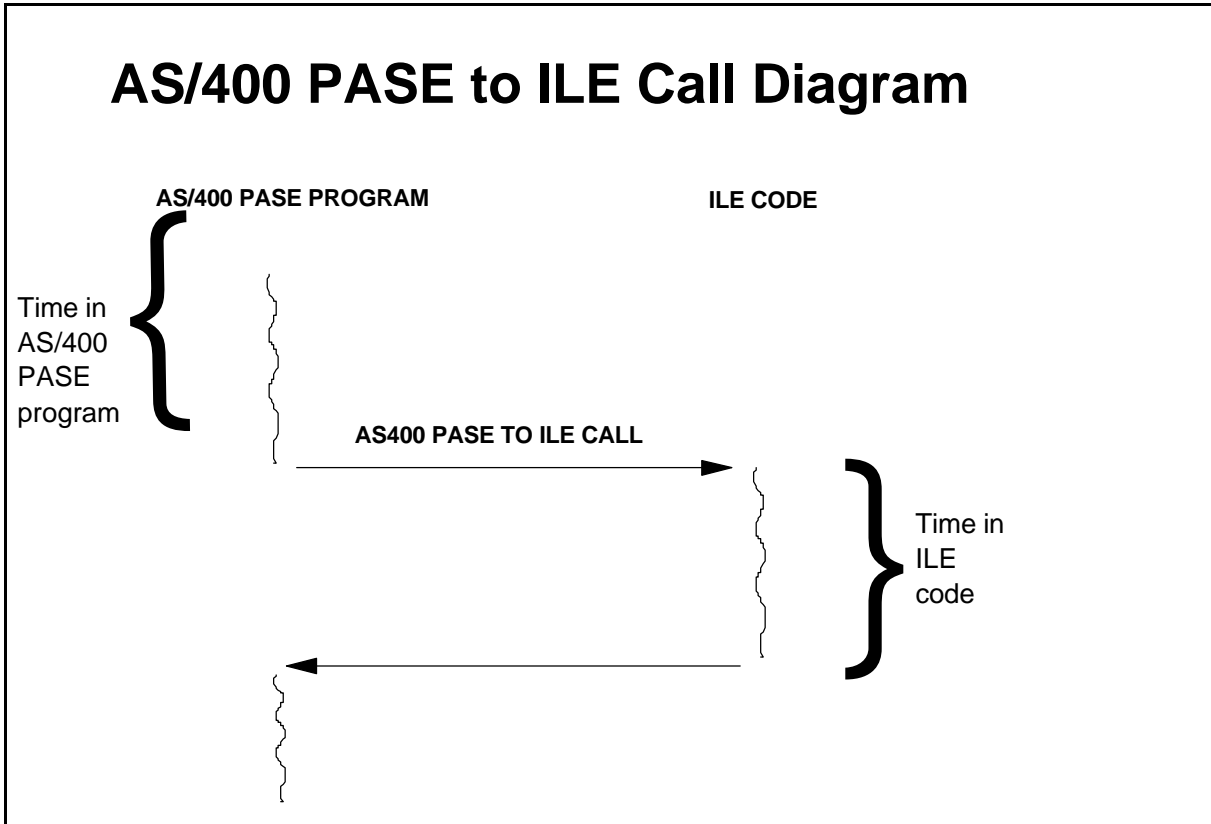
Figure 21.3 iSeries PASE to ILE Call Flow

DB2/400 CLI Performance Testing

iSeries PASE provides a shared library that enables access to the native iSeries 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 iSeries 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** Optimized level 40, inline	**AS/400 PASE** Optimized + inline	**%chg**
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 iSeries 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 iSeries 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 iSeries PASE CLI support is calling under the covers the native CLI support which is written in ILE, the results show how iSeries 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 iSeries 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 iSeries PASE

During the development of iSeries PASE, a number of commercial applications were ported to iSeries 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 iSeries 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 iSeries 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...).

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.

NetPerf Performance

This workload is a primitive-level function workload used to explore communication performance.

V5R1 vs V4R5 Netperf Measurements

No. of Sessions	transactions/sec		Instructions		CLT Cycles/transaction			SVR Cycles/transaction	
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.

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.

21.4 Summary

The performance comparisons of the three workloads (DB CLI, NetPerf and i2 benchmarks) that represent in the sections above show that we significantly improved the performance of OS/400 PASE in V5R1 compared to V4R5.

Chapter 22. IBM eServer Workload Estimator

22.1 Introduction

The purpose of the *IBM eServer Workload Estimator* 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 IBM Performance Management for eServer iSeries (PM eServer iSeries, previously 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 eServer iSeries data into the Estimator

If you are using PM eServer iSeries, you can easily merge your existing PM eServer iSeries data into the Estimator as a PM eServer iSeries 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 eServer iSeries 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 eServer iSeries workload to size the needed upgrade.

The PM eServer iSeries data is easily merged into the Estimator while viewing your PM eServer iSeries graphs on the web. To view your PM eServer iSeries graphs on the web, go to http://www.ibm.com/eserver/iseries/pm. Choose the 'click here to view your Management Summary Graph' button.

Follow these instructions to merge the PM eServer iSeries data into the Estimator:
1. Choose the 'Size my next upgrade' button.
2. Enter your PM eServer iSeries 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 eServer iSeries data is then passed into the Estimator. If this is your first time using PM eServer iSeries data with the Estimator, it is recommended that you take a few minutes to read the PM eServer iSeries integration with Workload Estimator tutorials. This text is found within the Estimator by clicking on Tutorials, followed by PM eServer iSeries.

If you are not familiar with PM eServer iSeries, refer to the iSeries Information Center at http://publib.boulder.ibm.com/iseries/v5r2/ic2924/index.htm

search for PM eServer iSeries or PM/400, or refer to the PM eServer iSeries web page at
http://www.ibm.com/eserver/iseries/pm for activation instructions.

22.3 Estimator Access

The *IBM eServer Workload Estimator* is available to everyone via the Internet at
http://www.ibm.com/eserver/iseries/support/estimator . Customers using the Estimator should review all
results with IBM or IBM Business Partner representatives.

New releases of the *IBM eServer Workload Estimator* 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 eServer Workload Estimator* 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://www.ibm.com/eserver/iseries/support/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://www.ibm.com/eserver/iseries/support/estimator
   - Once each day you will be shown the license agreement screen and are required to accept it.
2. Next you will see a "Workload Selection" screen
   - By default, the screen will show the available workloads that can be added to this estimation. Just
     click on one of the workloads and it will be added.
   - To add additional workloads, click on "add workload".
   - 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
- Options to control how the Estimator makes recommendations
  - The assumptions may be used once or stored as the defaults for the current Estimator user
  - Options 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 eServer Workload Estimator* 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 eServer Workload Estimator*.
  - 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 eServer Workload Estimator* 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 eServer Workload Estimator* provides, where it can be accessed, how it can be used with PM eServer iSeries (previously 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 eServer Workload Estimator*.

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 retrieval, single and multiple row insert, single row update, single row delete, journal, and commitment control. These operations are executed against files that vary from 100's of rows to 100's of millions of rows. Some files have multiple indexes, some only one. Some accesses are to the actual data and some take advantage of advanced functions such as index-only access.

A.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. Another aspect of CPW-type workloads is a dependency on a strong memory I/O subsystem. iSeries systems have always had a strong memory subsystem, but with the model 890, that subsystem was again significantly enhanced. Thus, CPW-like workloads see an additional benefit when moving to these systems.

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 Sizing and Performance Data Collection Tools

In this section, the following sizing and performance data collection tools are referenced.

- IBM eServer Workload Estimator (formerly know as IBM Workload Estimator for iSeries)

  The purpose of the IBM eServer Workload Estimator 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. *See Chapter 22 for a discussion on the IBM eServer Workload Estimator.*

- the AS/400 Capacity Planner (BEST/1 for the AS/400)

  Best for MES upgrade sizing, or complex 'new business' system sizing.

- the iSeries Batch Modeling tool STRBCHMDL. Formerly BATCH400.

  Best for MES upgrade sizing where the 'Batch Window' is important.

- Performance Data Collection Services

  This tool which is part of the OS/400 operating system collects system and job performance data which is the input for many of the sizing tools that are available today. It replaced the Performance Monitor in V5R1 and provides a more efficient and flexible way to collect performance data.

For more information on other iSeries Performance Tools, see the Performance Management web page at (http://www.ibm.com/servers/eserver/iseries/perfmgmt/).

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 withdrawn from the Performance Tools Licensed Product, effective with V5R2. No additional enhancements, including hardware table PTFs, will be available. Current customers of BEST/1 should consider alternative capacity planning tools. One possible alternative IBM tool is IBM Performance Management for eServer iSeries (PM eServer iSeries, or formally called PM/400) integrated with the IBM eServer Workload Estimator. 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 eServer iSeries.*

*Other alternative capacity planning products are available from other vendors. One product that will be available for V5R2 is **PATROL for iSeries - Predict** from BMC Software which will support the new i890*

and other current models. You will be able to find information about this product at
(http://www.bmc.com/products). Vendors such as Midrange Performance Group, CCSS Ltd and others
provide alternative products that can be used for capacity planning purposes.

What BEST/1 Does

The BEST/1 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 (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).

An IBM Global Campus course is available. The course is number S6095 - "iSeries Performance Analysis and Capacity Planning".

V5R1 Enhancements

No New Functional Enhancements: No new functional enhancements have been made to the V5R1 version of BEST/1. BEST/1 supports only the models that exist in the hardware table. Anything new will not be added to the hardware table. Refer to BMC's PATROL for iSeries - Predict product for newer models. The Patrol product will work with data created V4R4+ or collected data V4R2+ (i.e. convert to V4R4).

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).

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 utililization guidelines for each ASP in your model, 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.

B.2 Batch Modeling Tool (BCHMDL).   Formerly known as BATCH400.

BCHMDL 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.

BCHMDL is a tool to enable iSeries and AS/400 batch window analysis to be done using information collected by the OS/400 Collection Services.

BCHMDL 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).'

BCHMDL creates a 'model' from Collection Services performance data. This model will reside in a set of files named 'QAB4*' 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.

BCHMDL Output description:

1. Configuration summary shows the current and modeled hardware for DASD and CPU.

2. Job Statistics show the modeled results such as the following: Elapsed time, cpu seconds, cpu queuing seconds (how long the job waited for the processor due to it being in use by other jobs), disk seconds, disk queuing, exceptional wait time, cpu %busy, etc.

3. 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.

4. 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.

After looking at the results, use the change option to make changes to the processor, disk, or to the jobs themselves. You can increase the total workload by making copies of jobs or by increasing the amount of work done by any given job. If you have a long running single threaded job, you could model how fast it would run as 4 multithreaded jobs by making 4 copies but make each job do 1/4th the work.

For now this tool is on the IBM intranet at the following URL:

Http://ca-web.rchland.ibm.com/perform/perftool/batch400/batch400.htm

You will get a library named QBCHMDL. Add this library to your library list and start the tool by using the STRBCHMDL command.

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 an 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 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 eServer iSeries (formally called PM/400), PATROL for iSeries - Predict, 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 under the Systems Management topic in the V5R2 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 under the Systems Management topic in the V5R2 Information Center which is available at http://www.ibm.com/eserver/iseries/infocenter .

Appendix C. 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 processor intensive workloads such as Domino, SAP, and WebSphere. CIW values can be used in capacity planning analysis with the same cautions that were given above. **The IBM eServer Workload Estimator should be your first choice for sizing systems. See** *Appendix B,"* **iSeries Sizing and Performance Data Collection Tools" for more sizing information. After V5R2, CIW values will be discontinued.**

  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.

C.1 V5R2 Additions (February 2003)

New for this release is a product line refresh of the iSeries hardware which simplifies the model structure and minimizes the number of interactive choices. In most cases, the customer must choose between a Standard edition which includes a 5250 interactive CPW value of 0, or an Enterprise edition which supports the maximum 5250 OLTP capacity. The table in the following section lists the entire product line for 2003.

C.1.1 iSeries Model 8xx Servers

Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	5250 OLTP CPW*	Processor CIW*	MCU
Table C.1.1.1. iSeries Models 8xx Servers							
890-2498 (7427)	1300	1.41 MB*	24 - 32	29300-37400	Max	12900-16700	84100-108900
890-2498 (7425)	1300	1.41 MB*	24 - 32	29300-37400	0	12900-16700	84100-108900
890-2497 (7424)	1300	1.41 MB*	16 - 24	20000-29300	Max	8840-12900	57600-84100
890-2497 (7422)	1300	1.41 MB*	16 - 24	20000-29300	0	8840-12900	57600-84100
870-2486 (7421)	1300	1.41 MB*	8 - 16	11500-20000	Max	5280-9100	29600-57600
870-2486 (7419)	1300	1.41 MB*	8 - 16	11500-20000	0	5280-9100	29600-57600
825-2473 (7418)	1100	1.41 MB*	3 - 6	3600-6600	Max	1570-2890	8700-17400
825-2473 (7416)	1100	1.41 MB*	3 - 6	3600-6600	0	1570-2890	8700-17400
810-2469 (7430)	750	4 MB	2	2700	Max	950	7900
810-2469 (7428)	750	4 MB	2	2700	0	950	7900
810-2467 (7412)	750	4 MB	1	1470	Max	530	4200
810-2467 (7410)	750	4 MB	1	1470	0	530	4200
810-2466 (7409)	540	2 MB	1	1020	Max	380	3100
810-2466 (7407)	540	2 MB	1	1020	0	380	3100
800-2464 (7408)	540	2 MB	1	950	50	350	2900
800-2463 (7400)	540	0 MB	1	300	25	-	-

*Note: 1. 5250 OLTP CPW - Max (maximum CPW value). There is no limit on 5250 OLTP workloads
        and the full capacity of the server (Processor CPW) is available for 5250 OLTP work.
2. 1.41MB - These models share L2 cache between 2 processors
3. IBM does not intend to publish CIW ratings for iSeries after V5R2. It is recommended that
    the eServer Workload Estimator be used for sizing guidance, available at:
    http://www.ibm.com/eserver/iseries/support/estimator

C.1.2 Model 810 and 825 iSeries for Domino (February 2003)

Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	5250 OLTP CPW*	Processor CIW*	MCU
Table C.1.2.1. iSeries for Domino 8xx Servers							
825-2473 (7416)	1100	1.41 MB	6	6600	0	2890	17400
825-2473 (7416)	1100	1.41 MB	4	na	0	na	11600
810-2469 (7428)	750	4 MB	2	2700	0	950	7900
810-2467 (7410)	750	4 MB	1	1470	0	530	4200
810-2466 (7407)	540	2 MB	1	1020	0	380	3100

*Note: 1. 5250 OLTP CPW - With a rating of 0, adequate interactive processing is available for a
        single 5250 job to perform system administration functions.
2. IBM does not intend to publish CIW ratings for iSeries after V5R2. It is recommended that
   the eServer Workload Estimator be used for sizing guidance, available at:
   http://www.ibm.com/eserver/iseries/support/estimator
na - indicates the rating is not available for the 4-way processor configuration

C.2 V5R2 Additions

In V5R2 the following new iSeries models were introduced:
- 890 Base and Standard models
- 840 Base models
- 830 Base and Standard models

Base models represent server systems with "0" interactive capability. Standard Models represent systems that have interactive features available and also may have Capacity Upgrade on Demand Capability.

See Chapter 2, **iSeries RISC Server Model Performance Behavior,** for a description of the performance highlights of the new Dedicated servers for Domino models.

C.2.1 Base Models 8xx Servers

Table C.2.1.1 Base Models 8xx Servers							
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW	Processor CIW	MCU
890-0198 (none)	1300	1.41 MB*	32	37400	0	16700	108900
890-0197 (none)	1300	1.41 MB*	24	29300	0	12900	84100
840-0159 (none)	600	16 MB	24	20200	0	10950	77800
840-0158 (none)	600	16 MB	12	12000	0	5700	40500
830-0153 (none)	540	4 MB	8	7350	0	3220	20910

 * 890 Models share L2 cache between 2 processors

C.2.2 Standard Models 8xx Servers
Standard models have an initial offering of processor and interactive capacity with featured upgrades for activation of additional processors and increased interactive capacity. Processor features are offered through Capacity Upgrade on Demand, described in *C.3 V5R1 Additions*.

Table C.2.1.2. Standard Models 8xx Servers							
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW	Processor CIW	MCU
890-2488 (1576)	1300	1.41 MB*	24 - 32	29300-37400	120	12900-16700	84100-108900
890-2488 (1577)	1300	1.41 MB*	24 - 32	29300-37400	240	12900-16700	84100-108900
890-2488 (1578)	1300	1.41 MB*	24 - 32	29300-37400	560	12900-16700	84100-108900
890-2488 (1579)	1300	1.41 MB*	24 - 32	29300-37400	1050	12900-16700	84100-108900
890-2488 (1581)	1300	1.41 MB*	24 - 32	29300-37400	2000	12900-16700	84100-108900
890-2488 (1583)	1300	1.41 MB*	24 - 32	29300-37400	4550	12900-16700	84100-108900
890-2488 (1585)	1300	1.41 MB*	24 - 32	29300-37400	10000	12900-16700	84100-108900
890-2488 (1587)	1300	1.41 MB*	24 - 32	29300-37400	16500	12900-16700	84100-108900
890-2488 (1588)	1300	1.41 MB*	24 - 32	29300-37400	20200	12900-16700	84100-108900
890-2488 (1591)	1300	1.41 MB*	24 - 32	29300-37400	37400	12900-16700	84100-108900
890-2487 (1576)	1300	1.41 MB*	16 - 24	20000-29300	120	8840-12900	57600-84100
890-2487 (1577)	1300	1.41 MB*	16 - 24	20000-29300	240	8840-12900	57600-84100
890-2487 (1578)	1300	1.41 MB*	16 - 24	20000-29300	560	8840-12900	57600-84100
890-2487 (1579)	1300	1.41 MB*	16 - 24	20000-29300	1050	8840-12900	57600-84100
890-2487 (1581)	1300	1.41 MB*	16 - 24	20000-29300	2000	8840-12900	57600-84100
890-2487 (1583)	1300	1.41 MB*	16 - 24	20000-29300	4550	8840-12900	57600-84100
890-2487 (1585)	1300	1.41 MB*	16 - 24	20000-29300	10000	8840-12900	57600-84100

Table C.2.1.2. Standard Models 8xx Servers							
Model	Chip Speed MHz	L2 cache per CPU	CPU Range	Processor CPW	Interactive CPW	Processor CIW	MCU
890-2487 (1587)	1300	1.41 MB*	16 - 24	20000-29300	16500	8840-12900	57600-84100
890-2487 (1588)	1300	1.41 MB*	16 - 24	20000-29300	20200	8840-12900	57600-84100
830-2349 (1531)	540	4 MB	4 - 8	4200-7350	70	1630 - 3220	10680 - 20910
830-2349 (1532)	540	4 MB	4 - 8	4200-7350	120	1630 - 3220	10680 - 20910
830-2349 (1533	540	4 MB	4 - 8	4200-7350	240	1630 - 3220	10680 - 20910
830-2349 (1534)	540	4 MB	4 - 8	4200-7350	560	1630 - 3220	10680 - 20910
830-2349 (1535)	540	4 MB	4 - 8	4200-7350	1050	1630 - 3220	10680 - 20910
830-2349 (1536)	540	4 MB	4 - 8	4200-7350	2000	1630 - 3220	10680 - 20910
830-2349 (1537)	540	4 MB	4 - 8	4200-7350	4550	1630 - 3220	10680 - 20910

 * 890 Models share L2 cache between 2 processors

Other models available in V5R2 and listed in *C.3 V5R1 Additions* are as follows:
- All 270 Models
- All 820 Models
- Model 830-2400
- All 840 model listed in *Table C.3.4.1.1 V5R1 Capacity Upgrade on-demand Models*

C.3 V5R1 Additions

In V5R1 the following new iSeries models were introduced:
- 820 and 840 server models
- 270 server models
- 270 and 820 Dedicated servers for Domino
- 840 Capacity Upgrade on-demand models (including V4R5 models December 2000)

See Chapter 2, **iSeries RISC Server Model Performance Behavior,** for a description of the performance highlights of the new Dedicated Servers for Domino (DSD) models.

C.3.1 Model 8xx Servers

Table C.3.1.1 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
840-2461 (1543)	600	16 MB	24	20200	1050	10950	77800
840-2461 (1544)	600	16 MB	24	20200	2000	10950	77800

Table C.3.1.1 Model 8xx Servers							
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW	Processor CIW	MCU
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.

C.3.2 Model 2xx Servers

Table C.3.2.1 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

C.3.3 V5R1 Dedicated Server for Domino

Table C.3.3 .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

C.3.4 Capacity Upgrade on-demand Models

New in V4R5 (December 2000) , Capacity Upgrade on Demand (CUoD) capability offered for the iSeries Model 840 enables users to start small, then increase processing capacity without disrupting any of their current operations. To accomplish this, six processor features are available for the Model 840. These new processor features offer a Startup number of active processors; 8-way, 12-way or 18-way , with additional On-Demand processors capacity built-in (Standby). The customer can add capacity in increments of one processor (or more), up to the maximum number of On-Demand processors built into the Model 840. CUoD has significant value for installations who want to upgrade without disruption. To activate processors, the customer simply enters a unique activation code ("software key") at the server console (DST/SST screen).

The table below list the Capacity Upgrade on Demand features.

	Startup Processors ("Active")	On-Demand Processors ("Stand-by")	TOTAL Processors
840-2352 (2416)	8	4	12
840-2353 (2417)	12	6	18
840-2354 (2419)	18	6	24

Note: Features 23xx added in V5R1. Features 24xx were available in V4R5 (December 2000)

C.3.4.1 CPW Values and Interactive Features for CUoD Models

The following tables list only the processor CPW value for the Startup number of processors as well as a processor CPW value that represents the full capacity of the server for all processors active (Startup + On-Demand). Interpolation between these values can give an approximate rating for incremental processor improvements, although the incremental improvements will vary by workload and because earlier activations may take advantage of caching resources that are shared among processors.

Interactive Features are available for the Model 840 ordered with CUoD Processor Features. Interactive performance is limited by total capacity of the active processors . When ordering FC 1546, FC 1547, or FC 1548 one should consider that the full capacity of interactive is not available unless all of the On-Demand processors have been activated .For more information on Capacity Upgrade on-demand, see URL: : http://www-1.ibm.com/servers/eserver/iseries/hardware/ondemand

Note: In V5R2, CUoD features come with all standard models, which are described in the **V5R2 Additions** section of this appendix.

Table C.3.4.1.1 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 C.3.4.1.2 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

C.4 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.

C.4.1 AS/400e Model 8xx Servers

Table C.4.1 Model 8xx Servers (all new Condor models)					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	**Processor CPW**	**Interactive CPW**
820-2395 (1521)	400	n/a	1	**370**	**35**
820-2395 (1522)	400	n/a	1	**370**	**70**
820-2395 (1523)	400	n/a	1	**370**	**120**
820-2395 (1524)	400	n/a	1	**370**	**240**
820-2396 (1521)	450	2 MB	1	**950**	**35**
820-2396 (1522)	450	2 MB	1	**950**	**70**
820-2396 (1523)	450	2 MB	1	**950**	**120**
820-2396 (1524)	450	2 MB	1	**950**	**240**
820-2396 (1525)	450	2 MB	1	**950**	**560**
820-2397 (1521)	500	4 MB	2	**2000**	**35**
820-2397 (1522)	500	4 MB	2	**2000**	**70**
820-2397 (1523)	500	4 MB	2	**2000**	**120**
820-2397 (1524)	500	4 MB	2	**2000**	**240**
820-2397 (1525)	500	4 MB	2	**2000**	**560**
820-2397 (1526)	500	4 MB	2	**2000**	**1050**
820-2398 (1521)	500	4 MB	4	**3200**	**35**
820-2398 (1522)	500	4 MB	4	**3200**	**70**
820-2398 (1523)	500	4 MB	4	**3200**	**120**
820-2398 (1524)	500	4 MB	4	**3200**	**240**
820-2398 (1525)	500	4 MB	4	**3200**	**560**
820-2398 (1526)	500	4 MB	4	**3200**	**1050**
820-2398 (1527)	500	4 MB	4	**3200**	**2000**
830-2400 (1531)	400	2 MB	2	**1850**	**70**
830-2400 (1532)	400	2 MB	2	**1850**	**120**
830-2400 (1533)	400	2 MB	2	**1850**	**240**
830-2400 (1534)	400	2 MB	2	**1850**	**560**
830-2400 (1535)	400	2 MB	2	**1850**	**1050**
830-2402 (1531)	540	4 MB	4	**4200**	**70**
830-2402 (1532)	540	4 MB	4	**4200**	**120**

Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
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**

C.4.2 Model 2xx Servers

Table C.4.2.1 Model 2xx Servers					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW
250-2295	200	n/a	1	**50**	**15**
250-2296	200	n/a	1	**75**	**20**
270-2248 (1517)	400	n/a	1	**150**	**25**
270-2250 (1516)	400	n/a	1	**370**	**0**
270-2250 (1518)	400	n/a	1	**370**	**30**
270-2252 (1516)	450	2 MB	1	**950**	**0**
270-2252 (1519)	450	2 MB	1	**950**	**50**
270-2253 (1516)	450	4 MB	2	**2000**	**0**
270-2253 (1520)	450	4 MB	2	**2000**	**70**

C.4.3 Dedicated Server for Domino

Table C.4.3.1 Dedicated Server for Domino				
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Non Domino CPW	Interactive CPW
820-2425	450	2 MB	1	**100**	**0**
820-2426	500	4 MB	2	**200**	**0**
820-2427	500	4 MB	4	**300**	**0**
270-2422	400	n/a	1	**50**	**0**
270-2423	450	2 MB	1	**100**	**0**
270-2424	450	4 MB	2	**200**	**0**

C.4.4 SB Models

Table C.4.4.1 SB Models					
Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW*	Interactive CPW
SB2-2315	540	4 MB	8	7350	70
SB3-2316	500	8 MB	12	10000	120
SB3-2318	500	8 MB	24	16500	120

```
* Note: The "Processor CPW" values listed for the SB models are identical to the
830-2403-1531 (8-way), the 840-2418-1540 (12-way) and the 840-2420-1540 (24-way).
However, due to the limited disk and memory of the SB models, it would not be
possible to measure these values using the CPW workload. Disk space is not a high
priority for middle-tier servers performing CPU-intensive work because they are
always connected to another computer acting as the "database" server in a multi-tier
implementation.
```

C.5 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.

C.5.1 AS/400e Model 7xx Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

Table C.5.1.1 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

Model	Chip Speed MHz	L2 cache per CPU	CPUs	Processor CPW	Interactive CPW (Knee)	Interactive CPW (Max)
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
730-2067 (1510)	262	4 MB	4	2000	1050	1225
730-2067 (1511)	262	4 MB	4	2000	2000	2333.3
730-2068 (Base)	262	4 MB	8	2890	70	81.7
730-2068 (1508)	262	4 MB	8	2890	240	280
730-2068 (1509)	262	4 MB	8	2890	560	653.3
730-2068 (1510)	262	4 MB	8	2890	1050	1225
730-2068 (1511)	262	4 MB	8	2890	2000	2333.3
740-2069 (Base)	262	8 MB	8	3660	120	140
740-2069 (1510)	262	8 MB	8	3660	1050	1225
740-2069 (1511)	262	8 MB	8	3660	2000	2333.3
740-2069 (1512)	262	8 MB	8	3660	3660	4270
740-2070 (Base)	262	8 MB	12	4550	120	140
740-2070 (1510)	262	8 MB	12	4550	1050	1225
740-2070 (1511)	262	8 MB	12	4550	2000	2333.3
740-2070 (1512)	262	8 MB	12	4550	3660	4270
740-2070 (1513)	262	8 MB	12	4550	4550	5308.3

C.5.2 Model 170 Servers
Current 170 Servers

MAX Interactive CPW = Interactive CPW (Knee) * 7/6
CPU % used by Interactive @ Knee = Interactive CPW (Knee) / Processor CPW * 100
CPU % used by Processor   @ Knee = 100 - CPU % used by Interactive @ Knee
CPU % used by Interactive @ Max = Max Interactive CPW / Processor CPW * 100

Table C.5.2.1 Current Model 170 Servers									
Feature #	CPUs	Chip Speed	L2 cache per CPU	Processor CPW	Interactive CPW (Knee)	Interactive CPW (Max)	Processor CPU % @ Knee	Interactive CPU % @ Knee	Interactive CPU % @ Max
2289	1	200 MHz	n/a	50	15	17.5	70	30	35
2290	1	200 MHz	n/a	73	20	23.3	72.6	27.4	32
2291	1	200 MHz	n/a	115	25	29.2	78.3	21.7	25.4
2292	1	200 MHz	n/a	220	30	35	86.4	13.6	15.9

2385	1	252 MHz	4 MB	**460**	**50**	58.3	89.1	10.9	12.7
2386	1	252 MHz	4 MB	**460**	**70**	81.7	84.8	15.2	17.8
2388	2	255 MHz	4 MB	**1090**	**70**	81.7	92.3	6.4	7.5

Note: the CPU not used by the interactive workloads at their Max CPW is used by the system CFINTnn jobs. For example, for the 2386 model the interactive workloads use 17.8% of the CPU at their maximum and the CFINTnn jobs use the remaining 82.2%. The processor workloads use 0% CPU when the interactive workloads are using their maximum value.

AS/400e Dedicated Server for Domino

Table C.5.2.2 Dedicated Server for Domino									
Feature #	CPUs	Chip Speed	L2 cache per CPU	**Processor CPW**	**Interactive CPW**	Processor CPU% @ Knee	Processor CPU % @ Max	Interactive CPU % @ Knee	Interactive CPU % @ Max
2407	1	n/a	n/a	30	10	-	-	-	-
2408	1	n/a	4 MB	60	15	-	-	-	-
2409	2	n/a	4 MB	120	20	-	-	-	-

Previous Model 170 Servers

On previous Model 170's the knee of the curve is about 1/3 the maximum interactive CPW value.

Note that a constrained (c) CPW rating means the maximum memory or DASD configuration is the constraining factor, not the processor. An unconstrained (u) CPW rating means the processor is the first constrained resource.

Table C.5.2.3 Previous Model 170 Servers						
Feature #	Constrain / Unconstr	**Client / Server CPW**	**Interactive CPW (Max)**	Interactive CPW (Knee)	Interactive CPU % @ Max	Interactive CPU % @ Knee
2159	c	**73**	**16**	5.3	22.2	7.7
	u	**73**	**16**	5.3	22.2	7.7
2160	c	**114**	**23**	7.7	21.2	7.4
	u	**114**	**23**	7.7	21.2	7.4
2164	c	**125**	**29**	9.7	14	4.7
	u	**210**	**29**	9.7	14	4.7
2176	c	**125**	**40**	13.3	12.9	4.4
	u	**319**	**40**	13.3	12.9	4.4
2183	c	**125**	**67**	22.3	21.5	7.2
	u	**319**	**67**	22.3	21.5	7.2

C.6 AS/400e Model Sxx Servers

For AS/400e servers the knee of the curve is about 1/3 the maximum interactive CPW value.

Table C.6.1 AS/400e Servers							
Model	Feature #	CPUs	Max C/S CPW	Max Inter CPW	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
S10	2118	1	**45.4**	**16.2**	5.4	35.7	11.9
	2119	1	**73.1**	**24.4**	8.1	33.4	11.1
S20	2161	1	**113.8**	**31**	10.3	27.2	9.1
	2163	1	**210**	**35.8**	11.9	17	5.7
	2165	2	**464.3**	**49.7**	16.7	10.7	3.6
	2166	4	**759**	**56.9**	19.0	7.5	2.5
S30	2257	1	**319**	**51.5**	17.2	16.1	5.4
	2258	2	**583.3**	**64**	21.3	11	3.7
	2259	4	**998.6**	**64**	21.3	6.4	2.1
	2260	8	**1794**	**64**	21.3	3.6	1.2
S40	2207	8	**3660**	**120**	40	3.2	1.1
	2208	12	**4550**	**120**	40	2.6	0.8
	2256	8	**1794**	**64**	21.3	3.6	1.2
	2261	12	**2340**	**64**	21.3	2.7	0.9

C.7 AS/400e Custom Servers

For custom servers the knee of the curve is about 6/7 maximum interactive CPW value.

Table C.7.1 AS/400e Custom Servers							
Model	Feature #	CPUs	Max	Max	6/7 Max	CPU % @	CPU %
S20	2177	4	**759**	**110.7**	94.9	14.6	12.5
	2178	4	**759**	**221.4**	189.8	29.2	25.0
S30	2320	4	**998.6**	**215.1**	184.4	21.5	18.5
	2321	8	**1794**	**386.4**	331.2	21.5	18.5
	2322	8	**1794**	**579.6**	496.8	32.5	27.7
S40	2340	8	**3660**	**1050.0**	900.0	28.6	24.5
	2341	12	**4550**	**2050.0**	1757.1	38.6	33.1

C.8 AS/400 Advanced Servers

For AS/400 Advanced Servers the knee of the curve is about 1/3 the maximum interactive CPW value.

For releases prior to V4R1 the model 150 was constrained due to the memory capacity. With the larger capacity for V4R1, memory is no longer the limiting resource. In V4R1, the limit of 4 DASD devices is the constraining resource. For workloads that do not perform as many disk operations or don't require as much memory, the unconstrained CPW value may be more representative of the performance capabilities. An unconstrained CPW rating means the processor is the first constrained resource.

Table C.8.1 AS/400 Advanced Servers: V4R1 and V4R2								
Model	Feature #	Constrain / Unconstr	CPUs	Max C/S CPW	Max Inter CPW	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
150	2269	c	1	**20.2**	**13.8**	4.6	51.1	17
	2269	u	1	**27**	**13.8**	4.6	51.1	17
	2270	c	1	**20.2**	**20.2**	6.7	61.9	20.6
	2270	u	1	**35**	**20.6**	6.9	61.9	20.6
40S	2109	n/a	1	**27**	**9.4**	3.1	30.1	10
	2110	n/a	1	**35**	**14.5**	3.9	37.4	12.5
50S	2111	n/a	1	**63.0**	**21.6**	7.2	29.8	9.9
	2112	n/a	1	**91.0**	**32.2**	10.8	29.8	9.9
	2120	n/a	1	**81.6**	**22.5**	8.1	27.8	9.3
	2121	n/a	1	**111.5**	**32.2**	10.7	30	10
	2122	n/a	1	**138.0**	**32.2**	12.0	23.8	8.9
53S	2154	n/a	1	**188.2**	**32.2**	15.9	20.3	6.8
	2155	n/a	2	**319.0**	**32.2**	10.7	13.5	4.5
	2156	n/a	4	**598.0**	**32.2**	10.7	9	3
	2157	n/a	4	**650.0**	**32.2**	10.9	7.7	2.6

Table C.8.2 AS/400 Advanced Servers: V3R7								
Model	Feature #	Constrain / Unconstr	CPUs	Max C/S CPW	Max Inter CPW	1/3 Max Interact CPW	CPU % @ Max Interact	CPU % @ the Knee
150	2269	c	1	**10.9**	**10.9**	3.6	100.0	33.0
	2269	u	1	**10.9**	**10.9**	3.6	100.0	33.0
	2270	c	1	**27.0**	**13.8**	4.6	51.1	17.0
	2270	u	1	**33.3**	**20.6**	6.9	61.9	20.6
40S	2109	n/a	1	**27.0**	**9.4**	3.1	30.1	10
	2110	n/a	1	**33.3**	**13.8**	3.7	37.4	12.5
	2111	n/a	1	**59.8**	**20.6**	6.9	29.8	9.9
	2112	n/a	1	**87.3**	**30.7**	10.3	29.8	9.9
50S	2120	n/a	1	**77.7**	**21.4**	7.7	27.8	9.3
	2121	n/a	1	**104.2**	**30.7**	10.2	30	10
	2122	n/a	1	**130.7**	**30.7**	11.5	23.8	8.9
53S	2154	n/a	1	**162.7**	**30.7**	13.3	20.3	6.8
	2155	n/a	2	**278.8**	**30.7**	10.2	13.5	4.5
	2156	n/a	4	**459.3**	**30.7**	10.2	9	3
	2157	n/a	4	**509.9**	**30.7**	10.4	7.7	2.6

C.9 AS/400e Custom Application Server Model SB1

AS/400e application servers are particularly suited for environments with minimal database needs, minimal disk storage needs, lots of low-cost memory, high-speed connectivity to a database server, and minimal upgrade importance.

The throughput rates for Financial (FI) dialogsteps (ds) per hour may be used to size systems for customer orders. **Note: 1 SD ds = = 2.5 Fl ds**. (SD = Sales & Distribution).

Table C.9.1 AS/400e Custom Application Server Model SB1				
Model	CPUs	SAP Release	SD ds/hr @ 65% CPU Utilization	FI ds/hr @ 65% CPU Utilization
2312	8	3.1H	109,770.49	274,426.23
		4.0B	65,862.29	164,655.74
2313	12	3.1H	158,715.76	396,789.40
		4.0B	95,229.46	238,073.64

C.10 AS/400 Models 4xx, 5xx and 6xx Systems

Table C.10.1 AS/400 RISC Systems						
Model	Feature Code	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	V3R7 CPW	V4R1 CPW
400	2130	1	160	50	13.8	13.8
	2131	1	224	50	20.6	20.6
	2132	1	224	50	27	27
	2133	1	224	50	33.3	35
500	2140	1	768	652	21.4	21.4
	2141	1	768	652	30.7	30.7
	2142	1	1024	652	43.9	43.9
510	2143	1	1024	652	77.7	81.6
	2144	1	1024	652	104.2	111.5
530	2150	1	4096	996	131.1	148
	2151	1	4096	996	162.7	188.2
	2152	2	4096	996	278.8	319
	2153	4	4096	996	459.3	598
	2162	4	4096	996	509.9	650

Table C.10.2 AS/400e Systems					
Model	Feature Code	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	V4R3 CPW
600	2129	1	384	175.4	22.7
	2134	1	384	175.4	32.5
	2135	1	384	175.4	45.4
	2136	1	512	175.4	73.1
620	2175	1	1856	944.8	50
	2179	1	2048	944.8	85.6
	2180	1	2048	944.8	113.8
	2181	1	2048	944.8	210
	2182	2	4096	944.8	464.3
640	2237	1	16384	1340	319
	2238	2	8704	1340	583.3
	2239	4	16384	1340	998.6
650	2188	8	40960	2095.9	3660
	2189	12	40960	2095.9	4550
	2240	8	32768	2095.9	1794
	2243	12	32768	2095.9	2340

C.11 AS/400 CISC Model Capacities

Table C.11.1 AS/400 CISC Model: 9401

Model	Feature	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	CPW
P02	n/a	1	16	2.1	7.3
P03	2114	1	24	2.99	7.3
	2115	1	40	3.93	9.6
	2117	1	56	3.93	16.8

Table C.11.2 AS/400 CISC Model: 9402 Systems

Model	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	CPW
C04	1	12	1.3	3.1
C06	1	16	1.3	3.6
D02	1	16	1.2	3.8
D04	1	16	1.6	4.4
E02	1	24	2.0	4.5
D06	1	20	1.6	5.5
E04	1	24	4.0	5.5
F02	1	24	2.1	5.5
F04	1	24	4.1	7.3
E06	1	40	7.9	7.3
F06	1	40	8.2	9.6

Table C.11.3 AS/400 CISC Model: 9402 Servers

Feature Code	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	C/S CPW	Interactive CPW
S01	1	56	3.9	17.1	5.5
100	1	56	7.9	17.1	5.5

Table C.11.4 AS/400 CISC Model: 9404 Systems

Model	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	CPW
B10	1	16	1.9	2.9
C10	1	20	1.9	3.9
B20	1	28	3.8	5.1
C20	1	32	3.8	5.3
D10	1	32	4.8	5.3
C25	1	40	3.8	6.1
D20	1	40	4.8	6.8
E10	1	40	19.7	7.6
D25	1	64	6.4	9.7
F10	1	72	20.6	9.6
E20	1	72	19.7	9.7
F20	1	80	20.6	11.6
E25	1	80	19.7	11.8
F25	1	80	20.6	13.7

Table C.11.5 AS/400 CISC Model: 9404 Servers

Feature Code	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	C/S CPW	Interactive CPW
135	1	384	27.5	32.3	9.6
140	2	512	47.2	65.6	11.6

Table C.11.6 AS/400 CISC Model: 9406 Systems				
Model	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	CPW
B30	1	36	13.7	3.8
B35	1	40	13.7	4.6
B40	1	40	13.7	5.2
B45	1	40	13.7	6.5
D35	1	72	67.0	7.4
B50	1	48	27.4	9.3
E35	1	72	67.0	9.7
D45	1	80	67.0	10.8
D50	1	128	98.0	13.3
E45	1	80	67.0	13.8
F35	1	80	67.0	13.7
B60	1	96	54.8	15.1
F45	1	80	67.0	17.1
E50	1	128	98.0	18.1
B70	1	192	54.8	20.0
D60	1	192	146	23.9
F50	1	192	114	27.8
E60	1	192	146	28.1
D70	1	256	146	32.3
E70	1	256	146	39.2
F60	1	384	146	40.0
D80	2	384	256	56.6
F70	1	512	256	57.0
E80	2	512	256	69.4
E90	3	1024	256	96.7
F80	2	768	256	97.1
E95	4	1152	256	116.6
F90	3	1024	256	127.7
F95	4	1280	256	148.8
F97	4	1536	256	177.4

Table C.11.7 AS/400 Advanced Systems (CISC)					
Model	Feature Code	CPUs	Memory (MB) Maximum	Disk (GB) Maximum	CPW
200	2030	1	24	23.6	7.3
	2031	1	56	23.6	11.6
	2032	1	128	23.6	16.8
300	2040	1	72	117.4	11.6
	2041	1	80	117.4	16.8
	2042	1	160	117.4	21.1
310	2043	1	832	159.3	33.8
	2044	2	832	159.3	56.5
320	2050	1	1536	259.6	67.5
	2051	2	1536	259.6	120.3
	2052	4	1536	259.6	177.4

Table C.11.8 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