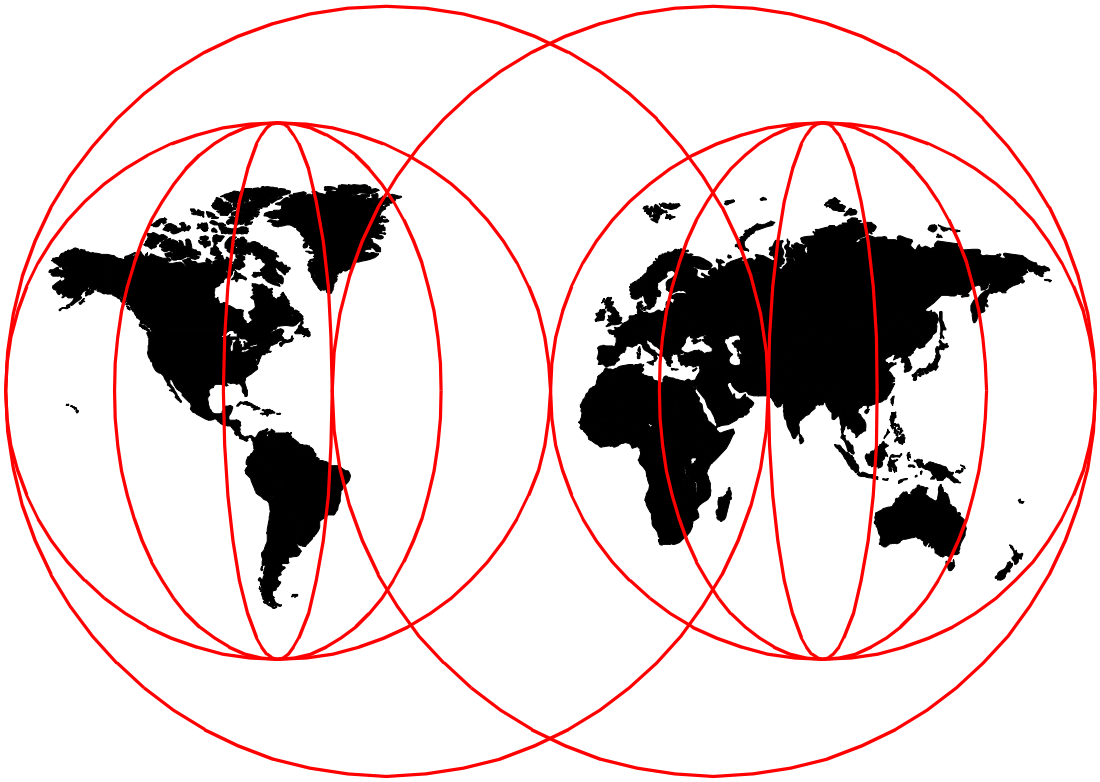
AS/400 HTTP Server Performance and Capacity Planning

Gottfried Schimunek, Erich Noriega, Greg Paswindar, George Weaver



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5645-00

International Technical Support Organization

SG24-5645-00

AS/400 HTTP Server Performance and Capacity Planning

January 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special notices” on page 195.

First Edition (January 2000)

This edition applies to Version 4 Release 4 of OS/400.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829

Contents

Figures

Tables

Preface

The Internet and Web browser-based applications have had a profound impact on how organizations distribute information, perform business processes, service customers, and reach new markets. At the heart of this is your HTTP server and Web-enabled data and applications. This redbook is intended for AS/400 programmers, network and system management professionals, and other information technologists that are responsible for designing, developing, and deploying Web-based applications and information systems. You will find a holistic approach to Web server and application performance for the key components of your solution.

This redbook discusses such key components as:

- The Web browser client
- Network and communications infrastructure
- Application characteristics, such as static or dynamically generated Web pages
- AS/400 server processor, memory, disk storage, and communications resources

The focus is on enabling you to create a sound methodology as a continual process for sizing and planning your Web application and HTTP server environment. This includes giving you the ability to:

- Measure the right data
- Correlate application transaction events, such as Web hits, to the measured data
- Analyze resource usage and transaction times

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Rochester Center.

Gottfried Schimunek is a certified IT Architect at the IBM International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of application development and performance. Before joining the ITSO in 1997, Gottfried worked in the Technical Support Center in Germany as a consultant for IBM Business Partners and customers, as well as for the IBM sales force.

Erich Noriega is a Systems Engineer with RISC (Reingenieria en Servicios de Computo), an IBM Premier Business Partner in Mexico City. Erich has extensive practical experience implementing new technologies on the AS/400 system, such as firewall and the Integrated Netfinity Server. He also has acted as an instructor on various courses at the IBM Educational Center in Mexico City covering TCP/IP and Internet technologies on the AS/400 system. He has eight years of experience with HTTP Servers on various platforms, and some of his main expertise areas include Personal Computer support, network design, Client Access, and AS/400 hardware and software.

Greg Paswindar is an AS/400 Specialist of the Business Server Group in IBM Indonesia. He has four years of experience in the AS/400 TCP/IP and performance areas. Prior to working at IBM, he was a senior system analyst. He holds a bachelors degree in Physics Engineering from Institut Teknologi Bandung (ITB). His areas of expertise include Internet, e-business, Client Access, and Netfinity.

George Weaver is an e-business consultant with the AS/400 Partners In Development organization at IBM Rochester, Minnesota. He has been with IBM for 17 years, with the last six in Partners In Development providing consultation, education, and writing to help customers and business partners enhance their AS/400 applications and services portfolios. He holds a bachelors and masters degree in Industrial Engineering. His current areas of expertise include AS/400 networking and communications, mail and messaging, HTTP server, directory, TCP protocols, and Domino for AS/400 topics.

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

Marcela Adan
Jim Cook
Thomas Gray
Marv Kulas
Kris Peterson
Ryan Rhodes
Jenifer Servais
Janet Willis
International Technical Support Organization, Rochester Center

Paul Albrecht
Jim Fall
Allan Johnson
Greg Olsen
Alexei Pytel
Bill Rapp
Saeid Sakhitabl
IBM Rochester

Mario Guerra Martinez
Carlos Zaldivar Perez
IBM Mexico

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Please send us your comments about this or other Redbooks in one of the following ways:

- Fax the evaluation form found in “IBM Redbooks evaluation” on page 207 to the fax number shown on the form.
- Use the online evaluation form found at: http://www.redbooks.ibm.com/
- Send your comments in an Internet note to: redbook@us.ibm.com

Chapter 1. AS/400 HTTP server performance overview

This chapter provides an overview of the AS/400 HTTP server that comes with OS/400 V4R4M0. The objective to give you a simple, concise introduction about the commonly used metrics for Web applications. In addition, you will find a methodology for Web application performance, along with some key concepts to help you understand how performance is measured on a Web-based application held inside an AS/400 system.

1.1 Overview

This chapter applies to customers with and without an AS/400 HTTP server. This section introduces you to the key concepts that can help you through this process.

1.1.1 Integrating different Web server components into one system

Traditionally, Web-based environments require separate boxes to generate a simple client dynamic response. Now, the AS/400 system allows you to integrate without spending long tedious hours trying to make everything work together, including applications, databases, HTTP servers, additional security measures, and so on. Figure 1 shows an example of an AS/400 Web-based transaction flow.

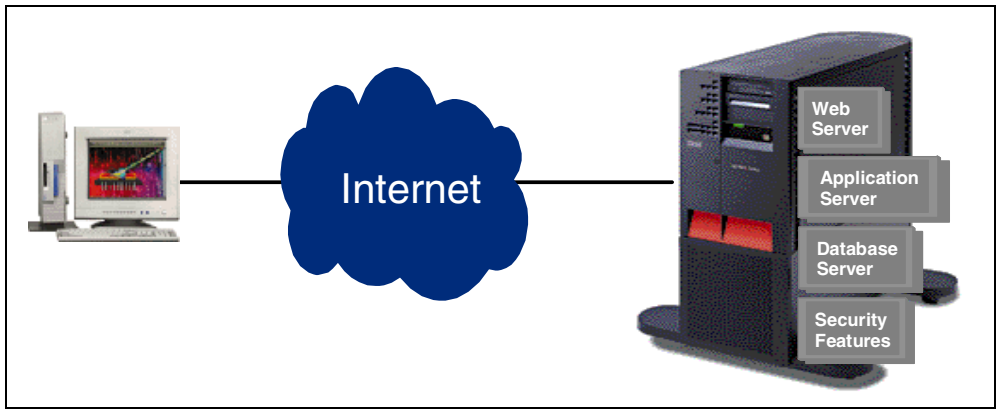


Figure 1. AS/400 Web-based transaction flow

How does all of this come together? The answer is simple. All of the benefits and features are built into the system. Tuning performance on your Web server becomes easier because you have all the technology in one central point from where you can tune every part that is involved in the complexity of Web transaction flows (Figure 2 on page 2).

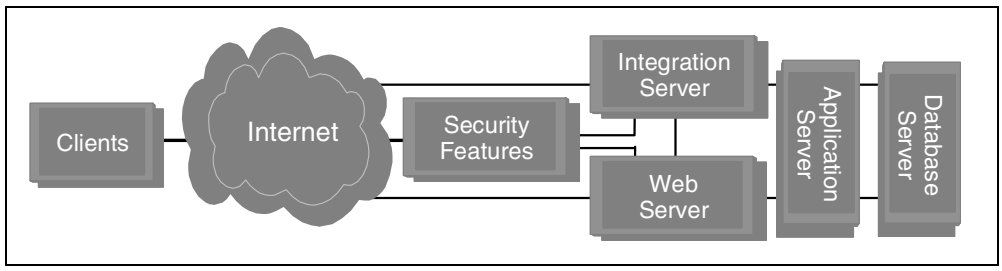


Figure 2. Traditional Web transaction flow

1.1.2 Understanding the components

In the short term, understanding the performance components of your system helps you to react quickly when a performance problem occurs at a crucial time. It may also help you know what you can expect from your system and, in some cases, from your environment. Capacity planning may help you understand (based on the information you previously collected, analyzed, extrapolated, and related) when this is going to happen to avoid getting dangerously close to peak capacity.

In the long term, you may need a more efficient system to prevent performance shortage problems from developing. You should also ensure that you have enough capacity on the system to handle unexpected workloads.

For example, suppose that you already set up an e-business site and that you are selling your products over the Internet. Since business tends to increase during the Christmas season, you need to ask whether your system is prepared to handle the additional demand. If your system is not prepared to manage the extra workload, you may lose many sales.

Since you are responsible for the systems on which your business runs, you know just how impatient users become while they are waiting for data when the pace increases. Their expectations are reduced to response time. For example, a user submits a form on a Web page that you administer. However, they may not be aware that to serve the "transaction completed" page, the data has to travel thousands of miles, run ten hundred lines of SQL, and access three different databases.

All of this takes us to the next statement: To tune performance on a Web-based environment, you should know every part of your system that is involved in this process. Maintaining good performance requires that you understand, plan, manage performance policies, and keep track of them. In other words, the tough work, when it comes to performance, is not accomplishing it, but maintaining it.

Stay focused, because you can spend a fair amount of time drilling down to get more details. This works well if you have the time, but it will only make you lose your focus. You should realize that, by concentrating on one objective, you can adversely affect another. You do not want to lose your performance objectives.

1.1.3 Tuning

Tuning Web-based applications is quite complex, especially when the Internet is involved. There are several factors that can be out of your control, such as traffic in the network, router capacity, and bottlenecks. Web application performance is

intimately related to application development. Your application must be developed in such a way that it has to handle every single resource it uses in an optimal way. If you did not consider performance for your application development, you may have problems trying to boost your response time. For example, the system and your application may run smoothly within your local area network, but taking your application to a Web-based environment has its own implications.

1.1.4 Components of Web environments

There are three major components of Web environments, each with its own performance requirements (Figure 3):

- Client
- Server
- Network

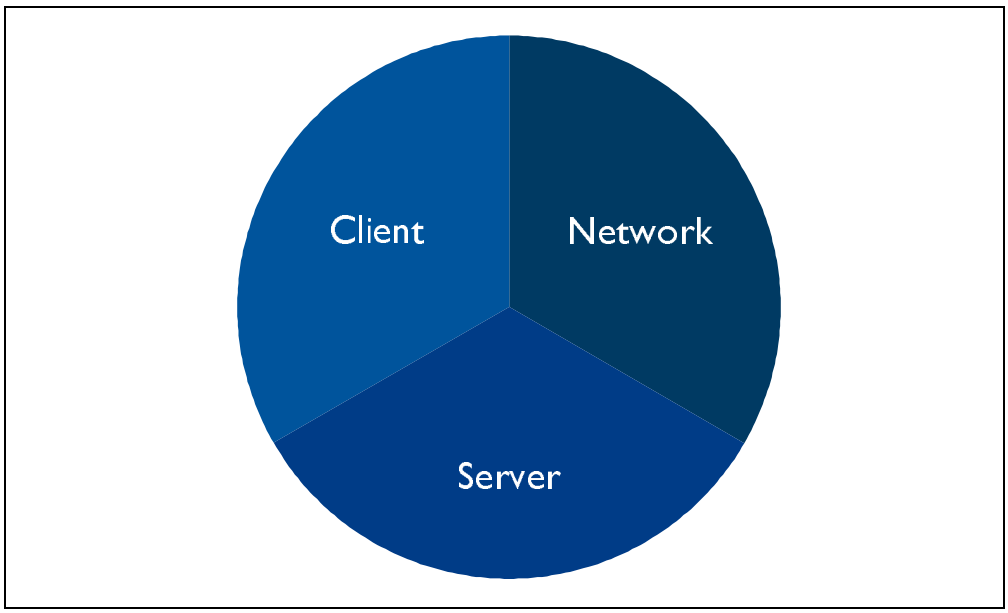


Figure 3. Response time components on Web environments

A problem in any of these areas may impact the overall performance. A problem may pertain to throughput, client processor speed, bandwidth, and resource utilization, to mention just a few factors that are involved in Internet scenarios.

1.1.4.1 Client

The client typically contributes up to 25% of the response time if you browse Web pages through a modem connection. If you access Web pages over a LAN or a WAN, this response time increases significantly, but the total response time decreases. In both cases, client performance relies on the following resources for which the client is responsible:

- **Processor speed**: Slower clients, such as a 486, 66Mhz desktop computer, may experience performance degradation compared to a Pentium client. We recommend that you upgrade your client processor to use faster process speeds on the client CPU.
- **Operating system**: Some operating systems have better performance in terms of Web browsing. Your operating system may be fast with

communications, but may not be fast enough when loading Java applets. Some operating systems have enhanced capabilities to modify parameters, for example, to deal with the network. Some operating systems allow you to increase the frame size you want to use for your network. Others do not let you do this. To minimize response time, consider using a different operating system if you think this is the reason for the delay in the response time. Typically, the operating system should not be a problem.

- **Memory**: Memory is an important factor inside the client since many Web-related tasks use large amounts of memory to be completed. For example, if you are trying to use cache on the clients, storing data in memory to retrieve it later is much faster than retrieving it from hard disk drives.
- **Hardware**: Every piece of hardware on your client is important. However, maybe your access speed for data on your hard drive is not fast enough, or your modem throughput is not enough when connecting to an ISP. To maximize utilization of these resources, you should upgrade your hardware. In some cases (as with some modems), you can upgrade your hardware drivers.
- **Browser**: Since Web browsers will become your main interface with the user, consider optimizing your browser on the clients. The available Web browsers on the market have several different strengths and weaknesses. For example, you may want to take advantage of cache capabilities that most of the browsers have. We recommend that you cache as close to the client when possible. This helps you to reduce client requests to the server.

1.1.4.2 Network

Usually the network piece has more impact on overall performance. The network usually contributes up to 60% of the total response time for either modem or LAN connections. This is because of a wide variety of factors such as throughput, network traffic, and speed of communication links. These factors can be examined in more detail if we look at such network components as:

- Routers
- LAN topology
- Link speed
- Modem throughput
- Packet filters
- Proxy
- Socks servers

You can spend a lot of time trying to get accurate network response times and never come up with an exact value. This is because the network is a dynamic environment from which you can only expect to retrieve average measurements from a measured period of time. Due to a lack of resources, and depending on the traffic outside your network, your company generally may not be able to go further. Nevertheless, you can act on your own network components and try to boost performance within your company realms. You may come to the conclusion that, to come up with smaller response times to your clients, you have to invest more in resources. This is a good conclusion, but you cannot do much about it. For example, you may need more bandwidth or an additional leased line.

1.1.4.3 Server

Now that you have reviewed two of the three major areas, we can go into the soul of this redbook: HTTP Server for AS/400 (License Program 5769-DG1). The purpose of this redbook is to provide a better perspective on how your AS/400

acts as a Web server. This redbook also describes and guides you through the process of understanding the implications of tuning your HTTP server. It is not our intention to teach you how to estimate the network or client capabilities, since every customer has different requirements and different scenarios. Nevertheless, we give you information on tools, hints, and tips on how to approach these subjects. We cover such topics as performance measurement and its analysis, performance characteristics of Web-based applications, capacity planning for Web-based applications, security performance impact, and how to do sizing on Web applications.

The server components are integrated from three major areas, including application, resources, and database. Figure 4 outlines the general composition of the server. However, you may find additional components on your own scenario. You need to fully understand and be aware of these components to attempt to acquire good performance.

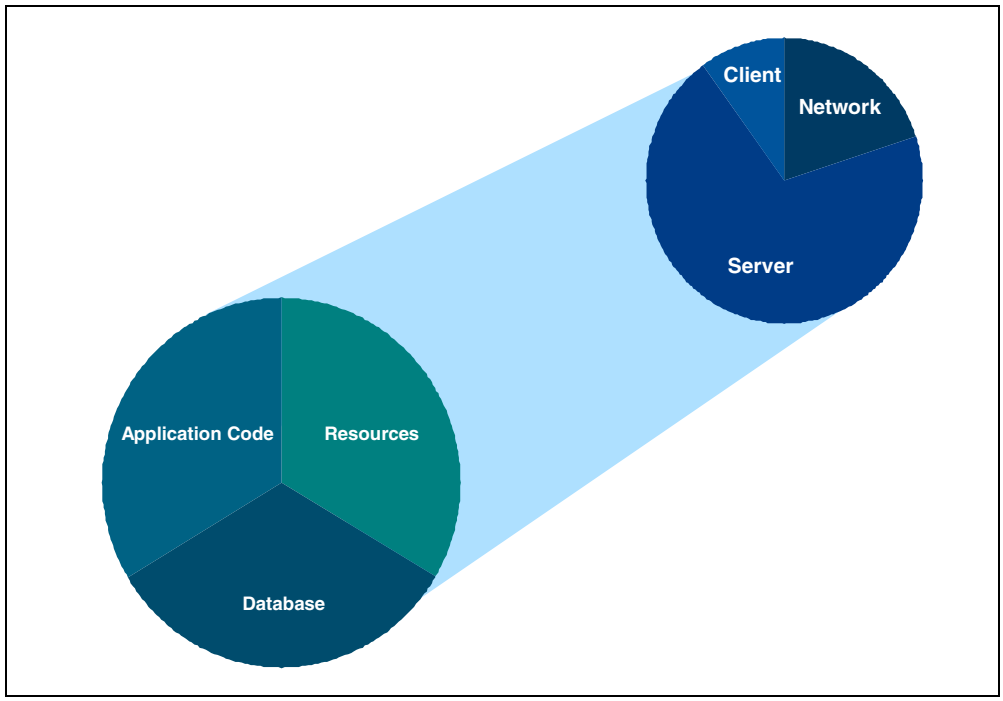


Figure 4. Server main components

Based on the objectives in this redbook, we can picture our performance analysis as shown in Figure 5 on page 6.

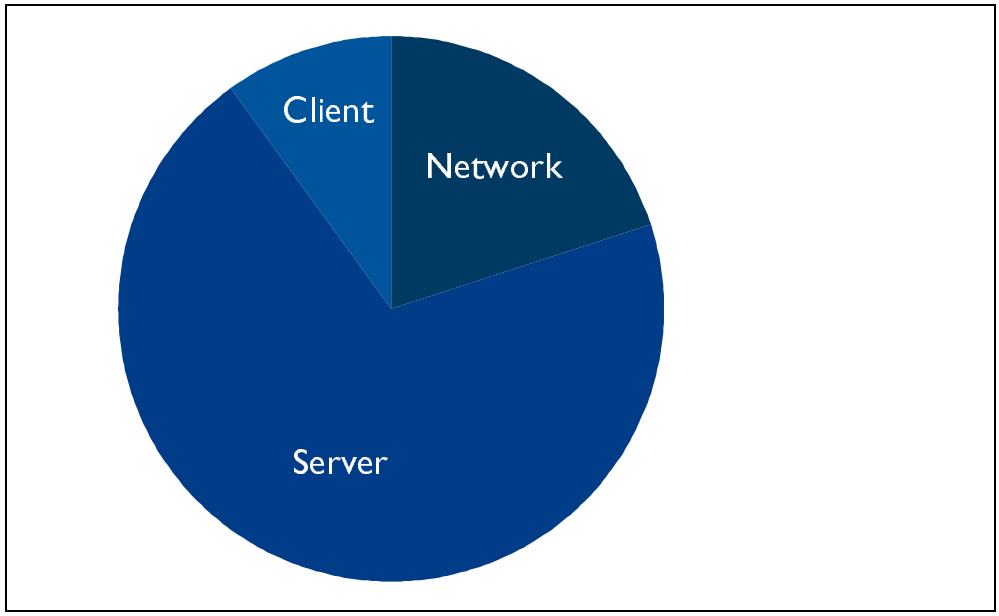


Figure 5. Web application response time components

Figure 6 shows how the sub-components are integrated into the server components. This may help you to see more clearly how performance is related to different "resources" inside your server and for which of those resources you should attempt to reduce response times.

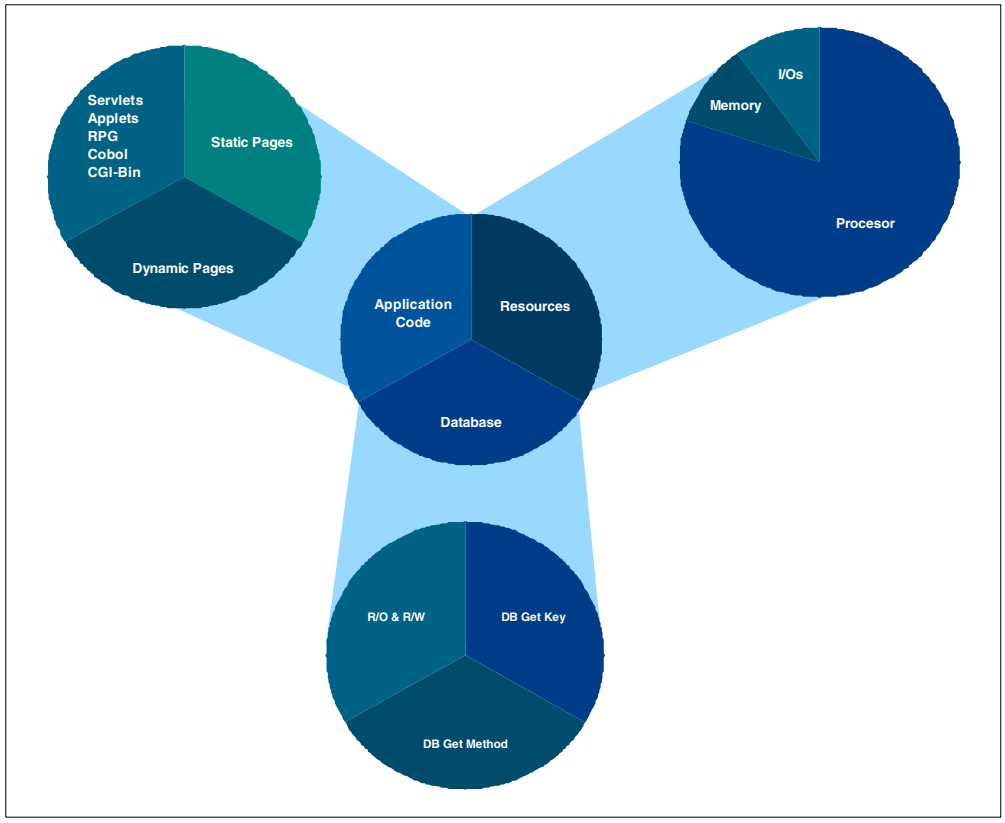


Figure 6. Example of some subcomponents

1.2 Basic queuing theory

Customer expectations for a single job or specific transaction must be balanced against realistic expectations when many jobs are active during the same time period. The work of a single job or transaction within that job is comprised of several tasks or services. The work given to a task is called a *request* or a *unit-of-work*. The task is also called a *server* and the time taken to complete processing the request is called the *service time*.

Note: A single server can service only one request at a time. Multiple requests wait for service by the server.

The servers of the components of response time include CPU time and disk I/O time. There are wait times associated with these servers, including waiting for CPU and waiting for disk I/O. These wait times are associated with queuing for the server. The higher the server utilization, the greater the wait or queuing time.

Queuing is a concept that applies to computer resources, just as it does to people waiting in line at the supermarket or waiting to access an Automated Teller Machine (ATM). In general, the time it takes to get a request or unit-of-work serviced, whether it is a request to complete the purchase at the supermarket counter, complete a cash withdrawal at the ATM, perform a disk I/O operation, or use the CPU, depends on three primary parameters:

- The number of "waiters" in the line ahead of a new request
- The number of servers responding to requests
- The service time to complete a request once it is given to the server, which is a function of the speed of the server and the amount of work to do

There are mathematical equations to determine the effect of queuing. Two of them, disk and CPU, are discussed in the following sections. The formula for computing the queuing multiplier assumes that:

- Work arrives in a normal (or Poisson) distribution pattern.
- Requests for the resources are not all for the same amount.

As the utilization of a server increases (more work for the server), queuing accounts for part of the longer work (or request) completion. In a Web transaction, this can be considered the major cause of long response times. The Queuing Multiplier (QM) is a measure of the effect of queuing. Using a simple example, assume that CPU is 67% utilized. The mathematical equation says that the QM for a single CPU is three. A QM of three means that, on the average, there are a total of three requests in the queue (you and two requests for work ahead of you). Therefore, using an average of .2 seconds of CPU to service a request, an interactive transaction (response time) takes a minimum of .5 seconds to use the CPU (server response time = QM x stand-alone service time). Refer to Figure 7 on page 8 for a graphical representation of the queuing effect.

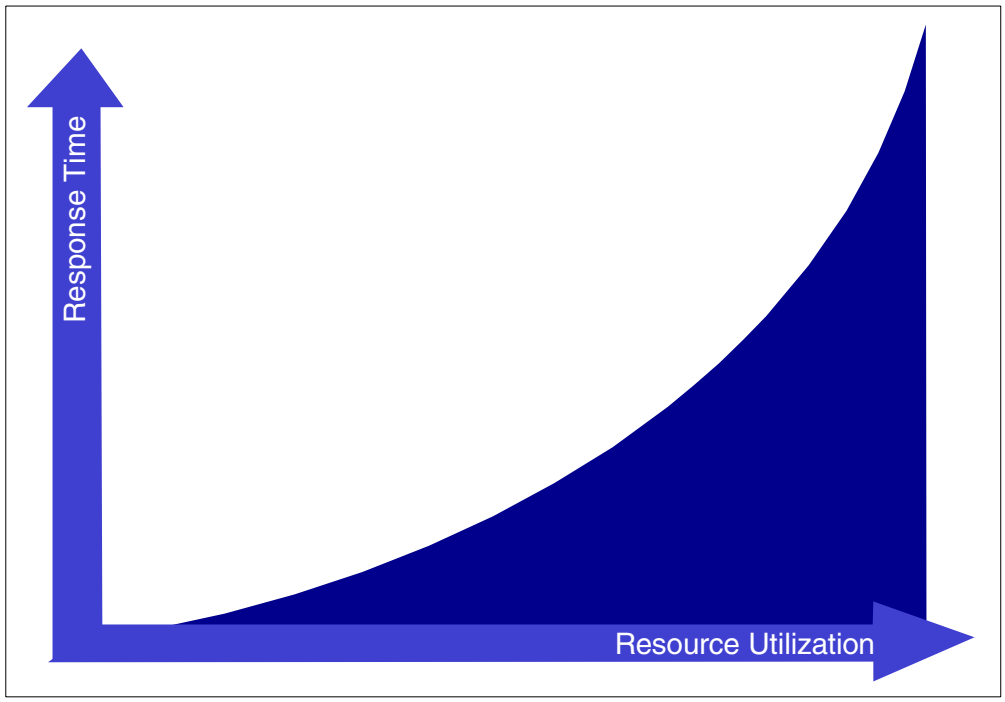


Figure 7. Response time versus resource utilization (queuing effect)

The components of response time show that CPU is only one of the resources (servers) involved in response time. Disk service time, which is a function of the disk utilization and the disk QM, also must be factored into response time expectations. In real environments, additional wait times, such as exceptional wait times, also need to be factored into expectations. These exceptional wait times (waiting for record or object locks, waiting for communication line data transmission, and so on) can play an important part in actual performance results and must be included in analyzing performance problems and capacity planning.

The Queuing Multiplier is an important factor when projecting the impact of adding work or additional hardware on current system performance. Note that the Performance Tools Capacity Planning support assumes a reasonably well-tuned system that assumes a CPU QM of four or less. Systems with performance problems often show resources with higher Queuing Multiplier factors. When using the Capacity Planner with measured data, a QM of greater than four generates less accurate results. The Performance Tools Transaction Report - Job Summary lists the CPU Queuing Multiplier calculated at each job priority for the collected data.

1.2.1 Queuing Multiplier effect

The Queuing Multiplier values used in the formulas for disk and CPU service time can be shown graphically. The curve on the graphic in Figure 8 shows the utilization at various rates and the significance of the "knee". The knee of the curve is the point where a change in utilization produces a corresponding higher change in the Queuing Multiplier. The change along the Y-axis (Queuing Multiplier) is significantly greater than the change along the X-axis (utilization). The knee of this curve is the maximum utilization point up to which a certain resource should be driven. After this "knee", service time becomes less stable and may increase dramatically for small utilization increases.

Not all resources (servers) have the same effect on performance at the same utilization values. There are different recommended maximum values for the different resources, such as CPU, disk, memory, controller, remote line, IOPs, and so on.

Performance Tools/400 V3R7, SC41-4340, provides more information on queueing. Figure 8 shows a simplified queuing formula and a curve derived from it. It highlights the effect of increasing utilization on the Queuing Multiplier for a single server. The results are based on the simple equation for single and normal random distribution:

```
(QM = 1/(1 - Utilization)
```

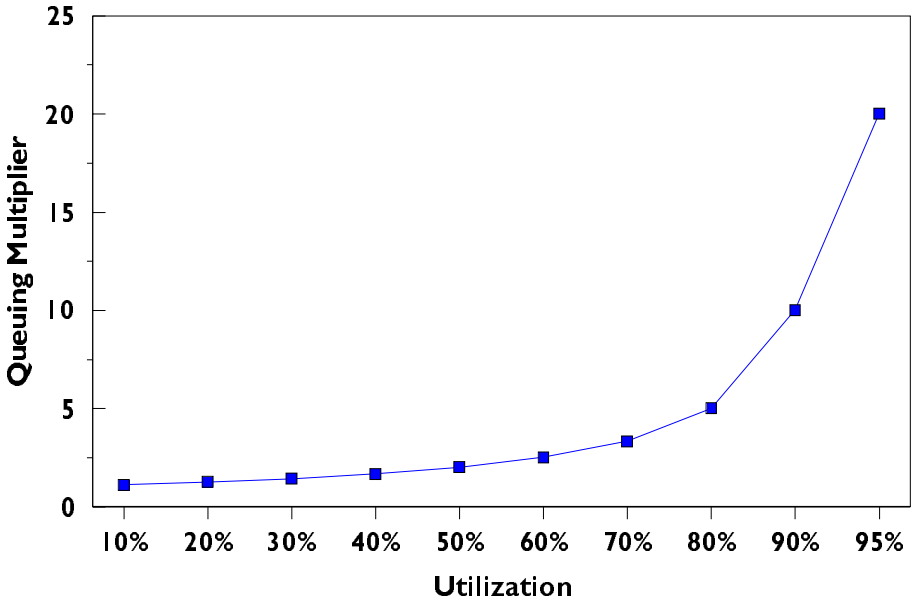


Figure 8. Queuing effect

1.2.2 Additional queuing considerations

There are a few additional considerations to be aware of when evaluating how queueing impacts performance. In particular, this section looks at the effects of multiple servers.

1.2.2.1 Multiple servers

The simple queuing theory equation discussed earlier assumes a single queue of requestors and a single server. In the high-end of the AS/400 product range, some models have multiple processors (N-way) that have more than one central processor executing instructions even though there is only a single queue of CPU requestors (Task Dispatch Queue). In this situation, the increased number of servers reduces the Queuing Multiplier and the average queue length. This scenario is shown in Figure 9 on page 10.

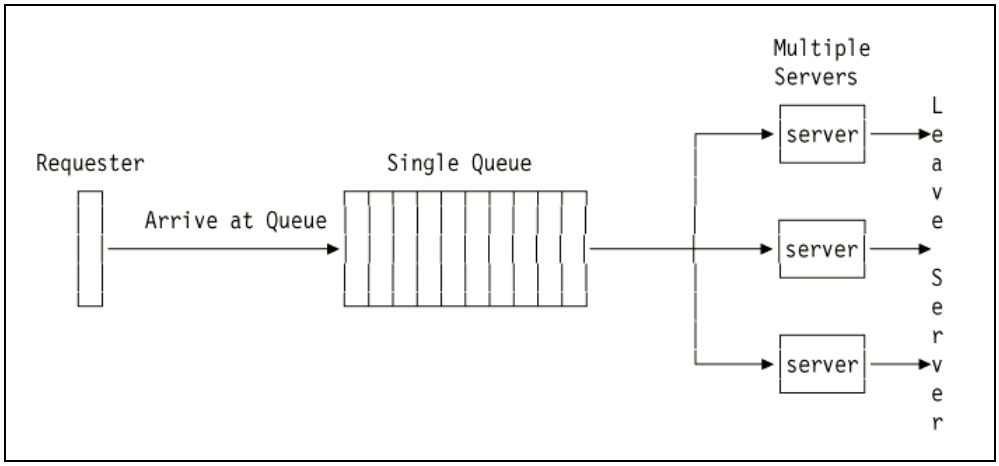


Figure 9. Single queue with multiple servers

Under these conditions, the Queuing Multiplier equation can be represented by a more specific form of the general "1/(1-U)" equation shown earlier:

```
QM = 1 / (1 - U**N)
```

In this equation, note the following points:

- `N` = Number of servers (processors)
- `U` = Utilization
- `**N` = To the Nth power

The graph in Figure 10 highlights the effect of multiple servers on QM.

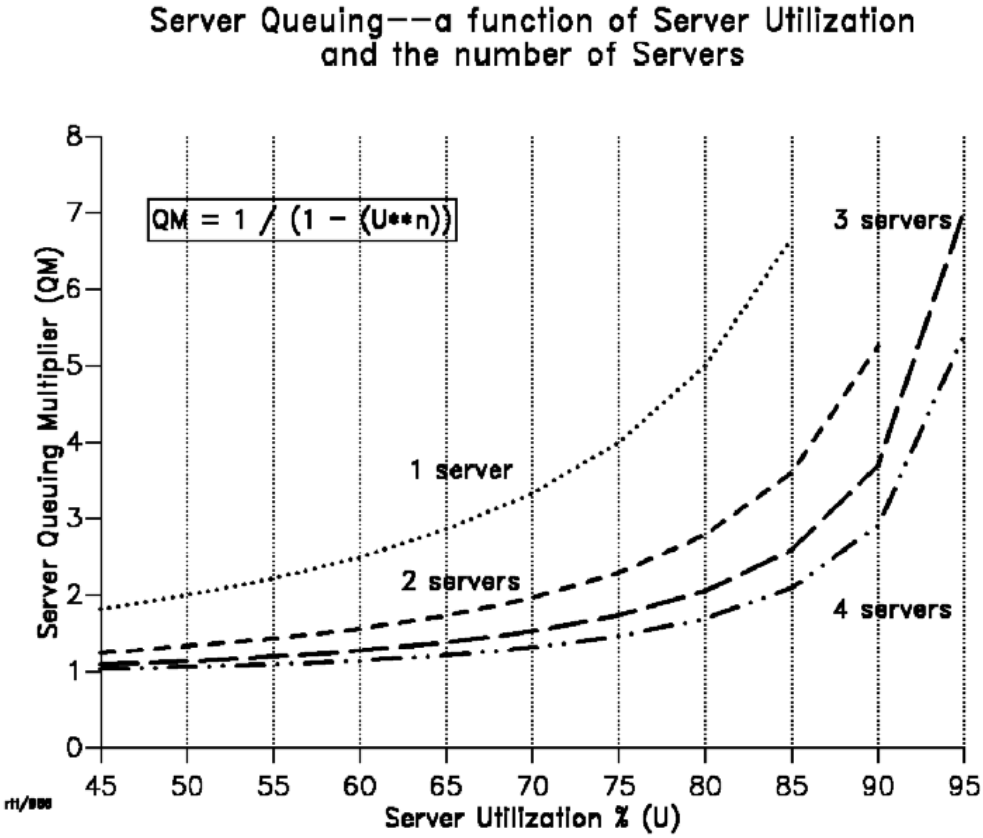


Figure 10. Queuing effect: Simple queuing equation for a multiple server

As the chart in Figure 10 shows, for a given Queuing Multiplier (QM), the QM value (value at that point in the QM versus Utilization graph) decreases as the number of servers increase. This means that for a given CPU utilization value, multiple server systems are less sensitive to increases in utilization than single server systems. The result of this is that you can operate the N-way CPUs at higher utilizations and maintain more stable response times.

The objective of setting and operating at or below the maximum resource utilization threshold guidelines is to minimize queuing and provide a stable environment for consistent response time.

1.2.2.2 Other factors that affect queuing

Two additional factors play an important part in calculating the Queuing Multiplier and predicting its effect. They are:

- Server utilization at equal and higher priority
- Number of requestors at equal priority

Use the following equation for more accurate CPU Queuing Multiplier computation:

$$\text{CPU QM} = \frac{1}{1 - \{(U1 + (U2 * (n-1)/n))\}^{**P}}$$

In this equation, note the following points:

- `U1` = CPU utilization for all higher priority jobs
- `U2` = CPU utilization for all equal priority jobs
- `n` = Number of competing jobs of equal priority
- `P` = Number of processors

This equation is the most detailed one used here for CPU Queuing Multiplier calculations. The simplified version 1/(1-U) can be derived from this equation.

The *performance metric* of an e-business solution is typically defined as the response time for the end user. The *performance problem* is often defined as "slow" response time when the user takes an action or series of actions while using a Web-based application.

1.3 Commonly used metrics for Web applications

In the different sections of this chapter, we refer to commonly used Web metrics, such as response time, transactions per second, or hits per second. Some of these metrics are directly related to the client and some others to the server. In fact, we can come up with a large list of metrics, such as the ones shown when you use a log analysis tool. But which ones concern you? What do you want to know about your server? How many details do you need to know? These are all simple questions. To get an answer for them, you need to ask more specific questions, such as:

- How may hits per day am I getting?
- Which are the most demanded pages on my Web site?
- How much bandwidth is my Web site using?

Your own questions will help you select the metrics that are useful for your business. How many hits per day am I getting? Or, if you want to dig a little deeper, ask yourself: How many hits per hour am I getting?

You can come up with a simple question: How many books am I selling? This simple question requires data achievement, analysis, and correlation to produce a simple answer.

There are few, but common, metrics. Basically, most of them are based on "hits". What is a hit? Every time you download a Web page, you may expect to hit a server once. This is not true. Every single object that is displayed inside your Web browser means a hit on the server (Figure 11). Notice that elements such as graphics, HTML code, or applets are hits on the server. If you are serving a Web page with seven .gif images, 1000 characters, and a Java applet, you can expect that every time this page is requested, you receive:

```
7 hits (one per image) + 1 hit (HTML Source CODE + 1000 Characters) + 1 hit
(initial request) = 9 hits per page
```

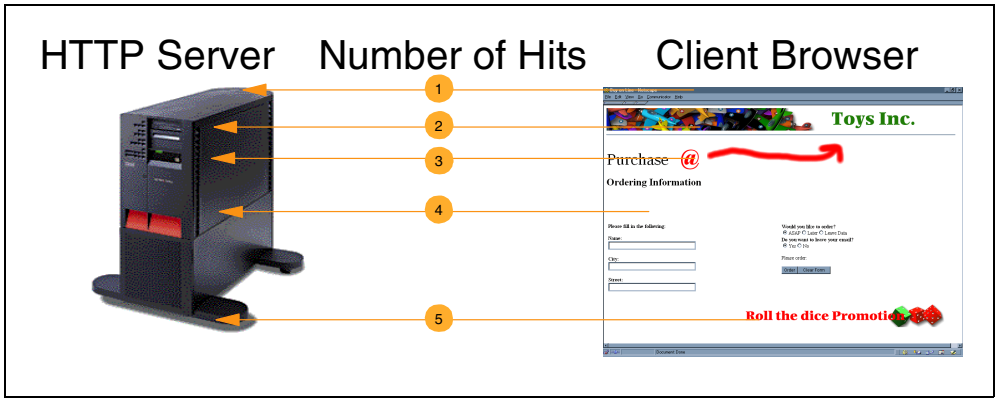


Figure 11. Each single object that is requested is considered a "hit"

There is a common metric on the AS/400 system related to measure the performance of the system as a Web server. This metric is Hits/Second/CPW. Please refer to the following section to review AS/400 Performance Metrics such as CPW.

1.4 AS/400 performance metrics

The performance capability of any given AS/400 system varies by model (B10, F97, S30, 740), release (V4Rx, V3Rx), workload type (interactive, batch, client/server), and many other complex factors. To provide a common metric that applies across the entire product line, the AS/400 system uses a set of relative performance measurements derived from commercial processing workload (CPW) on the AS/400 system. 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 that represented the maximum workload that can be applied to that model. This CPW value applied to either an interactive workload, a client/server workload, or a combination of the two.

Server models use two CPW values. The larger value represents the maximum workload that the model could support if it was entirely a client/server (for example, 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 was entirely interactive. For 7xx models, this is the CPW value for the interactive feature of the system.

An AS/400 HTTP server workload can be expressed in hits or transactions per unit of time, per CPW rating of the system. This varies with regard to the size of the objects, the type of request, and if security or encryption is involved.

1.5 Concepts of Web application performance

Web-based application performance can be a moderately to extremely complex subject. You may have little control over the end user's workstation and browser type, as well as the communications infrastructure (particularly if dealing with the Internet). Therefore, it is important to have a sound methodology in place for

planning your Web-based applications and, most importantly, to monitor results and make any adjustments based on user demands or growth requirements. The Web application performance planning cycle is shown in Figure 12.

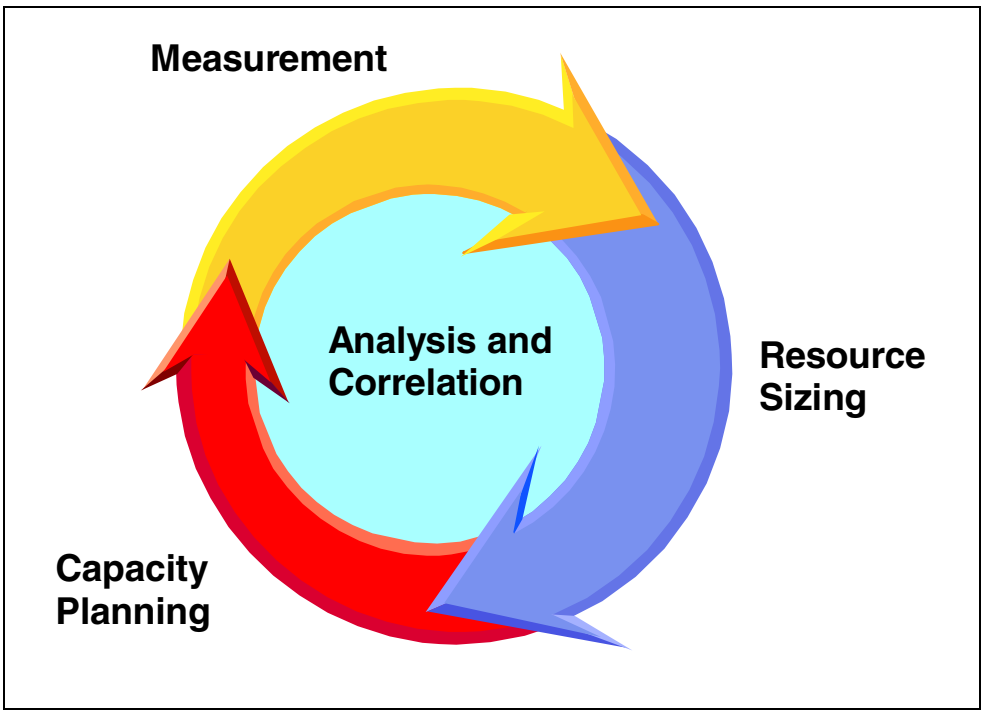


Figure 12. Web application performance planning cycle

1.5.1 Measurement

An old adage says, "If you can't measure it, you can't manage it." This is especially true in a Web application environment. If it takes a user 20 seconds to load your home page, how much of that time is due to the browser and workstation? How much is due to network traffic? How much is due to programming logic on our Web server? Similarly, we may know that our AS/400 HTTP server has a 10% CPU utilization average over a certain interval of time, but how does that relate to hits per second? A significant portion of this redbook deals with resource and transaction measurements.

The HTTP server provides data collection tools, such as logs, that record Web traffic through the server. Opening and analyzing the data logs can be done using a log analyzer tool. However, this tool can only give a glimpse of what is happening inside the system.

Performance Tools provide measurement, data analysis, and reporting. As part of Performance Manager, it is also used to help to create a strategy for planning, implementing, and controlling all tasks to measure and achieve acceptable performance, including capacity planning and collection of performance data. Figure 13 shows the measurement components.

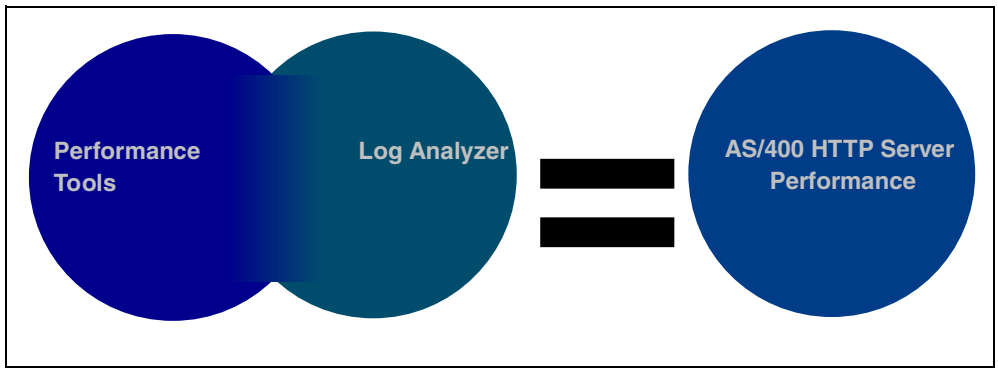


Figure 13. Measurement components

1.5.2 Analysis and correlation

Another critical phase is analysis. If you gather measurement data, you need to translate it into meaningful information. For example, you may want to determine how to correlate application activity such as a home page request or account inquiry on the client to resource usage such as network bandwidth, server CPU usage, or database access.

1.5.3 Sizing

Resource sizing entails estimating the service levels needed in your network, server, and client to satisfy a given set of workloads. Your prospective Web site will likely have a variety of application types, including static pages, interactive pages, and self-service type pages. You may want to enable secure business processes. Your users may want acceptable response time performance, regardless of the power of their workstation and any firewalls or gateways between them and the server.

1.5.4 Capacity planning

Last, but certainly not least, is the important task of capacity planning. Your sizing work gives you a best guess. The one you put on your Web site and online applications is when the work really begins. You may consider asking such questions as: What kind of traffic am I getting? What are the most common pages and requests? How are my server CPU, memory, DASD, and communications IOPs holding up? How much room for growth do I have before I need a bigger server, or a faster network connection?

The values in Figure 14 on page 16 were calculated based on the following information:

- Total Response Time = 13 seconds
- 13 seconds = 100 % Response Time
- Network Response Time = `(60*13)/100` = 7.8 seconds
- Client Response Time = `(15*13)/100` = 1.95 seconds
- Server Response Time = `(25*13)/100` = 3.25 seconds

Note the following equations:

```
13 Seconds - 100%
X^n - Z%
```

In these equations, note these points:

- X^n is the response time in seconds.
- Z% is the response time percentage.

```
7.8 sec (Network) + 1.95 sec (Client) + 3.25 sec (Server) = 13 sec (Total
Response Time)
```

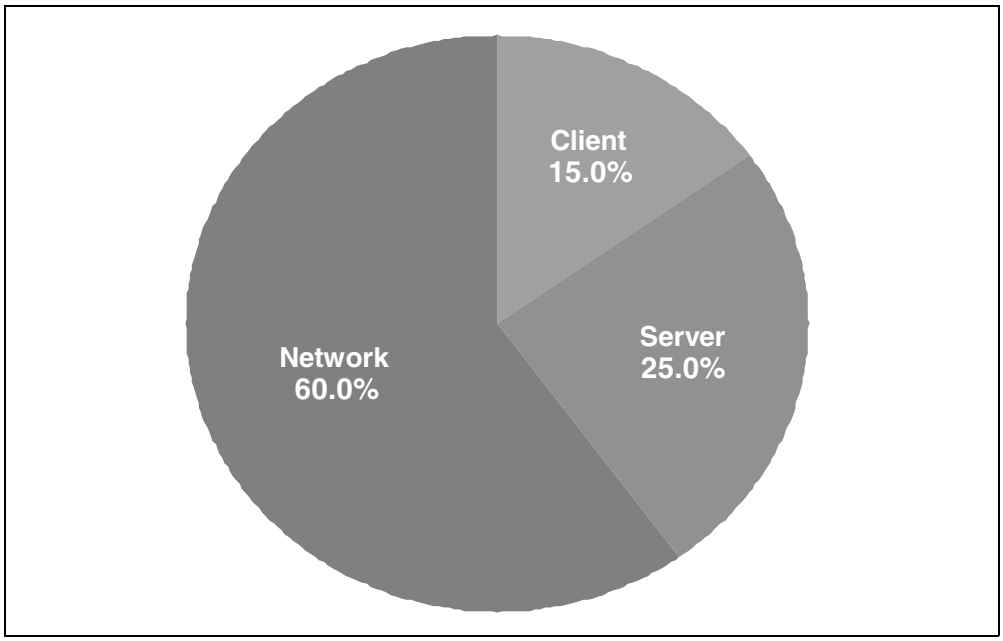


Figure 14. Example of estimating total response time

Note

Our equation simply shows example values. You should come up with your own values on response time.

1.5.4.1 Some facts

The HTTP server environment on RISC AS/400 models provides better performance per CPW than on CISC models. The HITS/SEC/CPW factor for RISC models is twice that available on CISC models. AS/400 server models may provide the best price/performance. Additionally OS/400 V4R4M0 has benefited from internal implementation improvements, including TCP protocol stack, especially over the 100 Mbps IOP and additional caching of Web pages frequently served.

1.6 The road to e-business

The Internet recently celebrated its 30th birthday, but its maturity has skyrocketed in the last four or five years since it opened up to commercial use. Internet Protocols (IP) have become the communications equivalent of "say what you mean." They are not just for Web browsing, but for other business applications, such as terminal-based 5250 applications or client/server applications written in an object-oriented programming language. You or your customers may already

have experienced what we term *exploitation* of this network computing, e-business evolution.

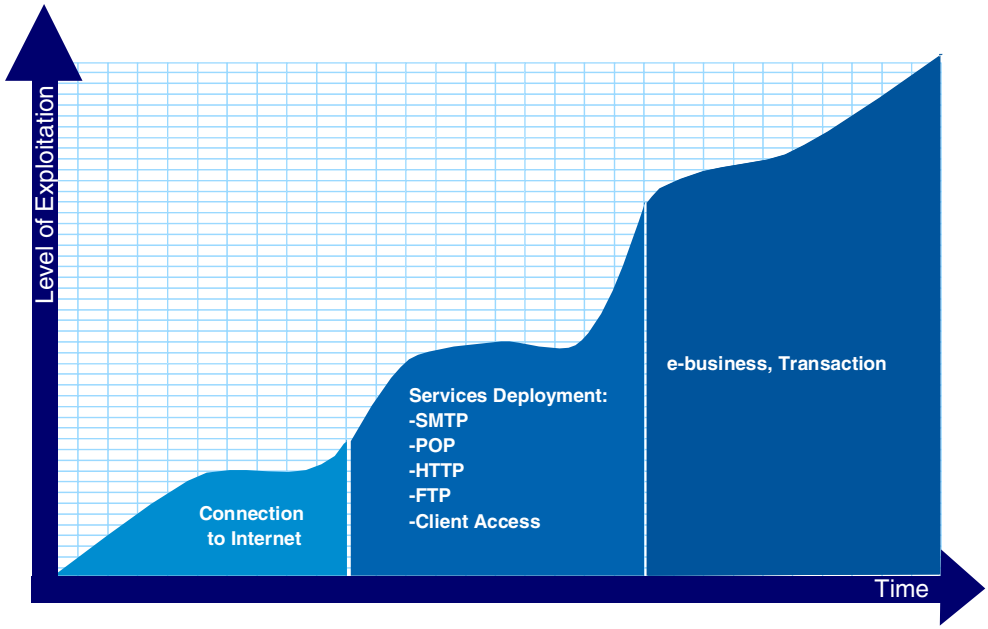


Figure 15. e-business road

The first phase is getting connected. This can be as simple as running your 5250-based applications over an IP network, or providing the network infrastructure between local and remote offices. It can grow to enabling Web access for your internal users and collaborative projects by exchanging mail over the Internet with colleagues, customers, or trading partners.

The second phase involves presence. This can be as simple as placing a basic home page on an Internet Service Provider's server. It can grow to hosting your own intranet Web site for a variety of internal information technology needs. It can grow to hosting your own external Web site, which you manage and maintain. It can grow into providing some level of customer self service, such as enabling your customers to look up account balances and request technical or marketing information on your products and services.

The next phase involves exploitation for competitive advantage. This enables your business to reach a new set of customers, regardless of where they are located or the time of day. It enables an efficient means of buying and selling goods and services. It also enables a more efficient means of conducting business than traditional brick and mortar facilities. And, most importantly, it is a *secure* means of conducting business.

This new, networked environment introduces many new complexities. We rely heavily on others to provide much of our infrastructure. The Web browser is becoming the universal client, and users can have their brand favorites, different versions, and maybe even device preferences. The network and communications infrastructure is even more variable. The IP family of protocols have great flexibility, but they generally do not have built-in class of service capabilities that we have been accustomed to with Systems Network Architecture (SNA).

Additionally, with the Internet, everyone and no one owns or is responsible for it. It is a collection of networks across the world, and service levels can vary greatly.

Finally, your server infrastructure must be adaptable. The AS/400 server has always been a world-class business application server for a variety of mission critical applications. As such, it fits in nicely with the networked world. However, the application characteristics are different than some of the traditional workloads, such as batch and interactive jobs. We need to be able to plan, implement, and manage in this new environment.

You may be reading this redbook because your business application is already running in a Web environment. Or, you may be reading this redbook instead of getting yourself or your entire team into the task of reviewing a large amount of documentation and trying to put all of the pieces together. If you are considering improving performance for your environment, this redbook is the best path to follow.

In some cases, performance just happens because you have either a group of performance experts tuning up your environment, or you seem to have enough resources. But, for how long would you have "enough resources"? If you are looking for a way to quickly learn and understand which issues are involved in Web-based environments regarding AS/400 systems, this redbook is definitely for you.

Chapter 2. Characteristics of Web-based applications

The Hypertext Transfer Protocol (HTTP) server allows an AS/400 system attached to a TCP/IP network to provide objects to any Web browser. At a high level, the connection is made, the request is received and processed, the data is sent to the browser, and the connection is terminated. HTTP server jobs and the communications router tasks are the primary jobs that should be involved and work tightly together to provide the service.

In OS/400 V4R4, the performance of the communications software infrastructure improved significantly based on socket, TCP/IP, and Ethernet performance. Many scenarios using this communications path reduced the CPU time required. Also, the scalability of communications performance improved due to the software enhancements to minimize contention. In addition, software enhancements for Ethernet support (with TCP/IP only) allow higher data rates and greater IOP capacities.

2.1 Basic Web serving

The basic protocol of Web serving is HTTP. It uses the Transmission Control Protocol (TCP) as a transport layer to retrieve the data. The HTTP protocol can be explained like this: The client establishes a connection to the server, and then issues a request. The server processes the request, returns a response, and closes the connection (Figure 16 on page 20).

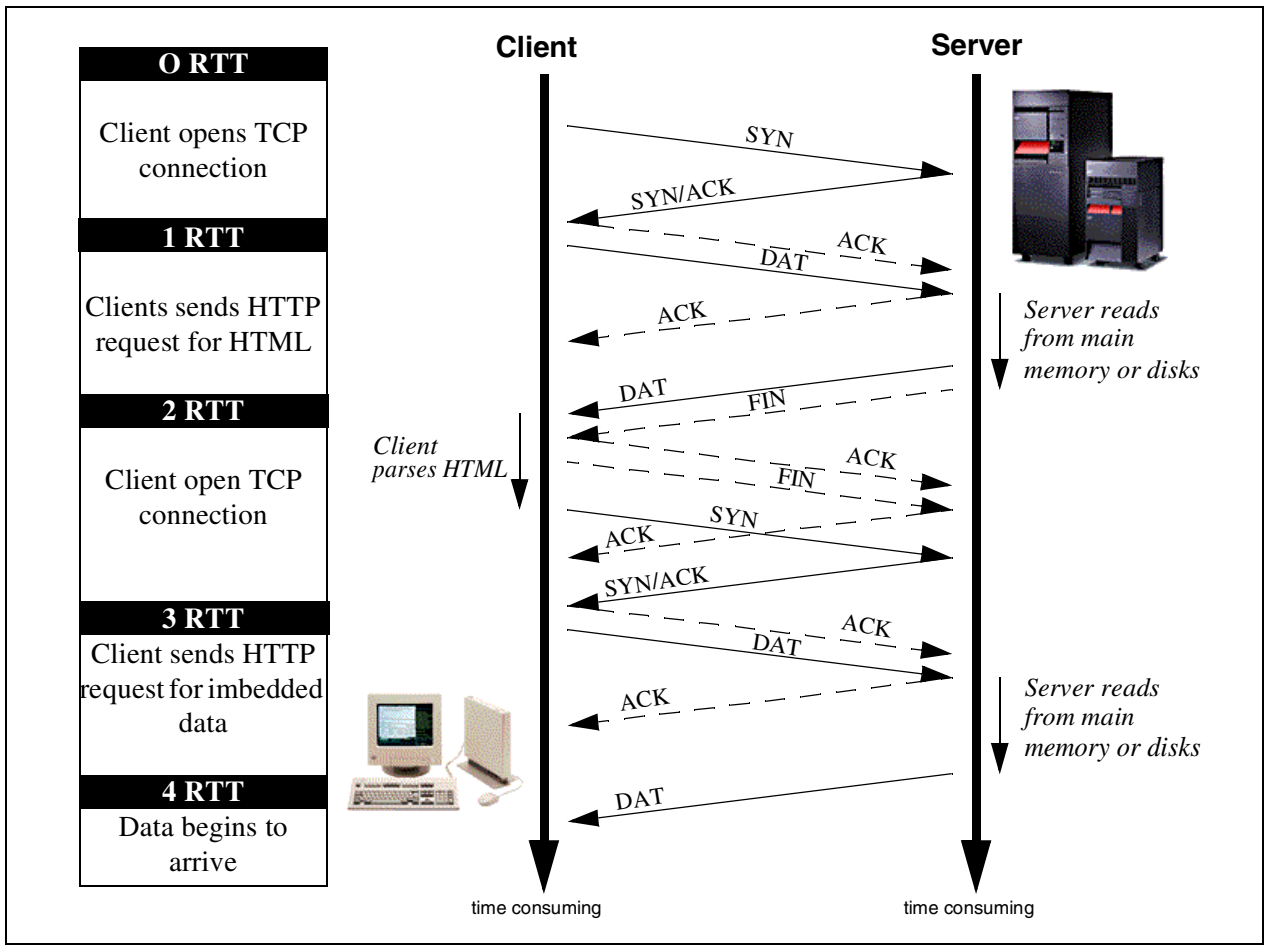


Figure 16. Illustration of an HTTP request

The actual traffic is more complicated, especially if the network path is included. The client asks the DNS first to get the IP address of the server, and then through a router to find the shortest path to the server. To simplify the model, the diagram in Figure 16 assumes that the client already knows the server.

Round-trips Time (RTT) is a measure of how long it takes for a packet to be sent from the client to the server and vice versa to complete the request. The vertical arrows shown in Figure 16 indicating server reads from disks and client parses HTML show the local delays. Each RTT is shown the client RTT required by HTTP protocol. Solid arrows either from the server to the client or from the client to the server are mandatory round-trips that result from packet exchanges. Dotted arrows are packets that are required by TCP protocol.

On each RTT, network path performance between the client and server affects the amount of time that is consumed.

2.2 Static page serving

HTTP is *stateless*, which means that the server or the client does not have provisions to track and record what has been done. HTTP is also connectionless, meaning that it has no persistent connections between the client and server. After

the server offers service to the request client, it has already given service to another client or another request.

A static page is connectionless (HTTP/1.1). After giving service to a client, the HTTP server does not maintain any network resources. Although it is connectionless, due to the TCP works, the protocol remains active to make sure that all packets were sent successfully. Setting the TCP keep alive definition to "long time" improves the service, but it uses a lot of network resources.

HTTP is also asymmetric. The number of request bytes from the client is much smaller than the reply bytes from the server. In Figure 16, the total request packet and sent packet most likely have the same amount, because the client needs to send ACK back to the server to acknowledge the sent packet from the server. When this happens, the server becomes a bottleneck.

Another consideration is matching the size between the static page and the router packet. If a Web page size is bigger than the maximum router packet size can transfer at one time, the router cuts down the size so its maximum capacity can be handled. There is an advantage if the Web size is equal to, or less than, the router capacity.

2.3 Dynamic page serving

Dynamic Web pages typically require script programming, which produces the actual HTML content sent to the Web browser. This may be required for HTML form handling, database access, or the delivery of specific content based on the browser type requesting the information. This may also include scripts that run within the Web browser to handle HTML form validation, multimedia, and special effects (excluding auto-generated scripts produced by Web authoring tools).

You are no longer constrained to use sequential content laid out linearly in your Web pages. By specifying positions for blocks of HTML content, you can decide what content goes where on the page, instead of leaving it up to the browser to lay it out for you. Using JavaScript, for example, you can change the layout of your page dynamically, and you can modify the page in a variety of ways after the user has opened it. You can make content vanish or appear, and you can change the color of individual parts of your page. You can incorporate animation into your Web pages by moving and modifying individual parts of your HTML page "on the fly."

2.4 Interactive application page serving

Interactive sessions generate lots of interruptions, for example, with each mouse movement or each keystroke. Such interruptions disturb the performance of another Web user. Every single interactive movement or keystroke is then sent to the server and creates most traffic in the network.

Examples of interactive Web pages include:

- Flash
- Shockwave for Director
- Java applets
- JavaScript

If you give the Web server a lower priority, the AS/400 system responds to another process first. After the higher priority process finishes, the next priority is given to the Web server. To overcome this, if your Web applications are interactive, give the Web server a higher priority. The default priority of a Web server and another batch type job is 25. The default priority for an interactive job is 20.

If the operating system, kernel, and HTTP daemon receive the HTTP request, they are processed as shown in Figure 17. When an AS/400 Web server receives an HTTP request, it must schedule some CPU time for the httpd daemon to handle the request (Figure 17). Httpd runs as a user process, so it has a lower priority than a kernel process. It must wait and share the remaining time with other user processes.

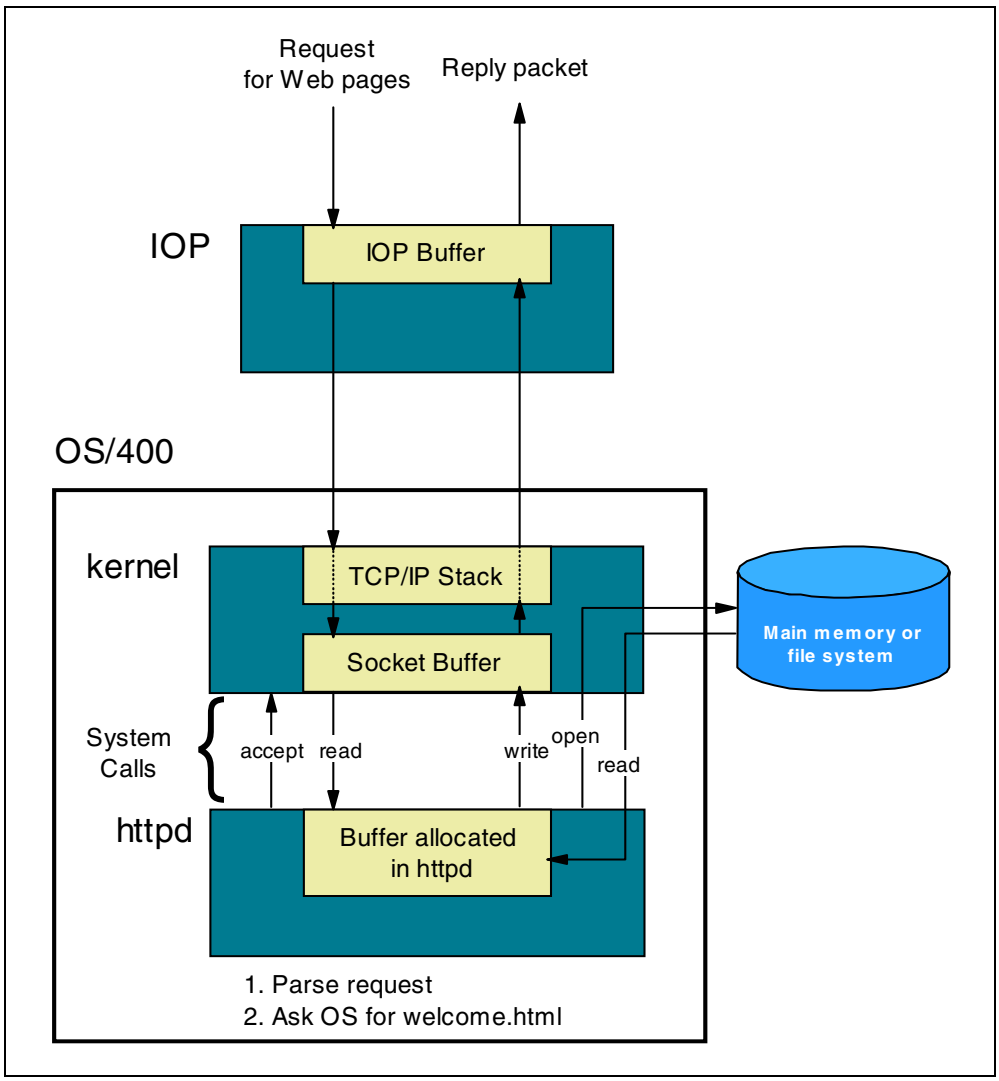


Figure 17. OS and httpd service the request

For good interactive applications, you can separate the Web server job to another server and give the Web server the higher priority. The default security model for Java applets allow access only back to the Web server from which the applets originated. This means that the Web server must not only serve the applet, but also deal with requests from that applet to service database access.

To change the job priority, you can create your own Web server job and use the different priority of your new Web server job.

2.5 Persistent connection

A significant difference between HTTP/1.1 and earlier versions of HTTP is that persistent connections are the default behavior of any HTTP connection. That is, unless otherwise indicated, the client should assume that the server will maintain a persistent connection, even after error responses from the server.

Persistent connections provide a mechanism by which a client and a server can signal the close of a TCP connection. This signaling takes place using the Connection header field. Once a close is signaled, the client must not send any more requests on that connection.

Prior to persistent connections, a separate TCP connection was established to retrieve each URL, which increased the load on HTTP servers and caused congestion on the Internet. The use of inline images and other associated data often requires a client to make multiple requests of the same server in a short amount of time.

Persistent HTTP connections have a number of advantages, such as:

- By opening and closing fewer TCP connections, CPU time is saved in routers and hosts (clients, servers, proxies, gateways, tunnels, or caches), and memory used for TCP protocol control blocks can be saved in hosts.
- HTTP requests and responses can be pipelined on a connection. Pipelining allows a client to make multiple requests without waiting for each response, allowing a single TCP connection to be used much more efficiently, with much lower elapsed time.
- Network congestion is reduced by reducing the number of packets caused by TCP opens, and by allowing TCP sufficient time to determine the congestion state of the network.
- Latency on subsequent requests is reduced, since there is no time spent in TCPs connection opening handshake.

Clients using future versions of HTTP may optimistically try a new feature. However, if communicating with an older server, they should retry with old semantics after an error is reported.

2.6 Internet, intranet, and extranet deployment

Intranet users have the advantage of using available bandwidth more efficiently with guaranteed service. A Web server that opens to the Internet should have the best connection. If it uses the same bandwidth with heavy traffic intranet users, Internet users will likely starve to get the service. A designer may use a different machine for the intranet Web server and Internet Web server, but use the same contents for replication. Intranet Web servers are located inside the firewall, and Internet Web servers are located outside the firewall.

Other designers use the same Web server for both the Internet and intranet, but they are separated by different network cards. One network card follows the

blueprint of the intranet, with an intranet IP address and is connected behind the firewall. Another network card uses an IP address from the Internet Service Provider (ISP) and connects outside the firewall.

Figure 18 shows the attributes of the different nets.

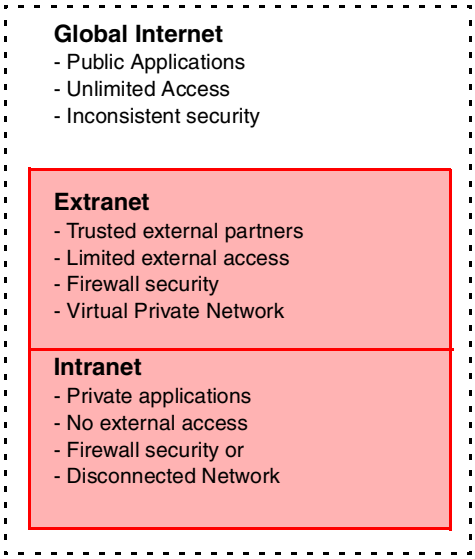


Figure 18. Internet, intranet, and extranet

Internet users may only be allowed to get HTTP and NNTP service. Internet users, facing unpredictable network performance and lower bandwidth, also get latency of service due to many routers that hopped from the client to the server. Intranet users gain all TCP services, such as HTTP, NNTP, SMTP, LPD, POP3, DNS, etc.

The key characteristic of an extranet is that protected applications are not visible, because of the firewall or other security mechanism between two or more connected organizations to global Internet users. Extranets must be protected from the global Internet and must also share the Internet characteristic that applications and services are not visible to the larger Internet.

2.7 Integrated commercial applications

A commercial application is usually integrated between Web applications and backend applications (such as ERP applications). Such an application directs a Web server to call an external program and access a large database. In some cases, the Web server opens a socket to a separate machine that is running the database and leaves it open. Some "middleware" products, such as CICS, manage a single open connection to a database.

There are three types of database access:

- **Static database access**, such as Net.Question, for searching between Web servers
- **Data mining** for marketing purposes
- **Transactional database**, such as from the buyer to the merchant and to the credit card issuer

Database planning and tuning requires more attention compared to Web server planning and tuning. Some databases are given HTTP servers to eliminate the layer between the client and the database. They can convert the data to HTML pages (such as 5250-HTTP gateway) "on the fly" and maintain the state of transactions. They can open one connection to all database requests, instead of creating one connection per request.

A typical commercial application to invoke a Web server and database is described in Figure 19.

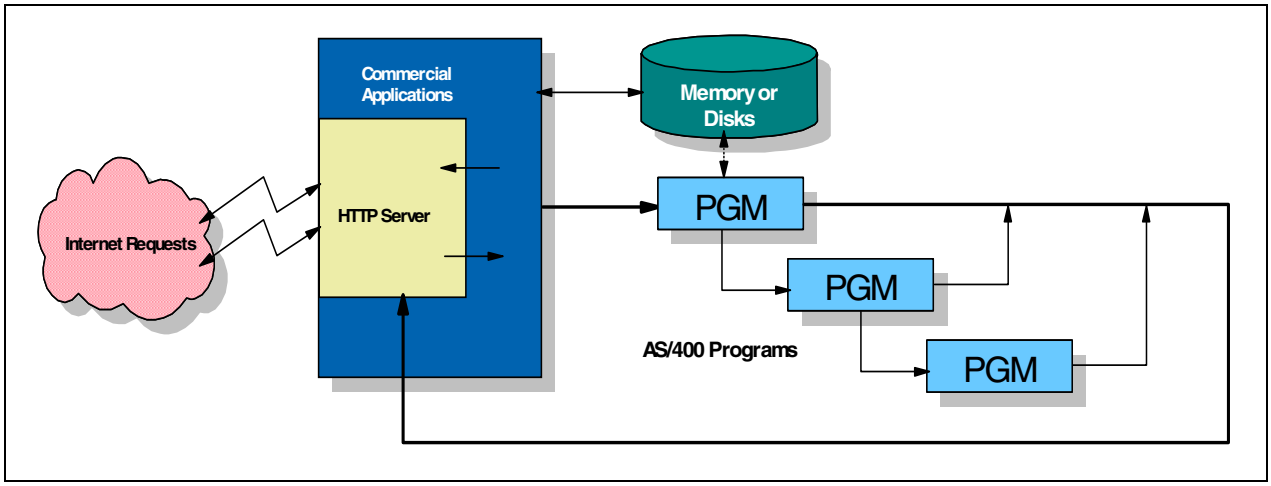


Figure 19. Typical commercial application access to AS/400 programs

IBM Net.Commerce is an example of IBMs commercial applications integrated with a Web server. Net.Commerce works together with the relational database and the secure Web server. For more information, refer to 8.4, “Net.Commerce applications” on page 184.

Chapter 3. Performance measurement and analysis tools

To analyze the Web load, use the following tools and applications:

- AS/400 Performance Tools Reports
- Web Access Log Analysis Tool

AS/400 Performance Reports uses the System Report and the Component Report to examine overall AS/400 activity and its load.

The *Web Access Log Analysis Tool* uses the built-in analysis tools and Web data mining in OS/400 V4R4. It can be combined with available tools in the marketplace for observing the behavior of Web users.

3.1 AS/400 data

There is a variety of software that enables the AS/400 system to function effectively and efficiently as an information server for Internet and intranet users. In addition to coding applications efficiently, the server environment must be configured for optimum performance. In the *AS/400 Performance Capabilities Reference Guide*, SC41-0607, and other documents, there are a number of tips to optimize the AS/400 TCP/IP communications capabilities.

From the AS/400 Performance license program, there are three parameters that are the most important for observing the AS/400 Web server:

- CPU utilization
- Communication IOP/LAN utilization
- Main memory and disk arm

For detailed information about AS/400 system performance matters, please refer to *AS/400 Performance Management V3R6/V3R7*, SG24-4735.

In OS/400 V4R4, you can choose which collector data performance to use. There are two choices, and both provide essentially the same data:

- Use Collection Services from within AS/400 Operation Navigator.
- Use Performance Monitor.

As the AS/400 Operation Navigator and Collection Services evolve, future support for performance data collection may focus on Collection Services. Collection Services allows you to gather performance data with little or no observable impact on system performance.

For more details on installing and setting up Collection Services, refer to:
`http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html`

See Management Central under the Operation Navigator topic. Or, you can access the Information Center from AS/400e Information Center CD-ROM. For an introduction to Collection Services, read Appendix A of *Work Management*, SC41-5306.

3.1.1 CPU utilization

Performance Tools does not support Web-specific data. In other words, the HTTP server does not collect or report any specific data to any of the tools. Since Web

server is a job with multiple threads, you can find a lot of traditional statistics in QAPMJOBxxx files.

All QAPMJOBxxx files that are needed can be viewed by using Print Performance Report. Performance Tools pulls data from the various performance monitor databases files to create the performance reports. These databases are:

- QAPMJOBS
- QAPMJOBMI
- QAPMBML

The QAPMJOBS file is the Performance Monitor database file, which contains job-related performance statistics. In V4R4, Performance Monitor was re-engineered. As a result, it was produced as *Collection Services*. Basically, Collection Services is a new way to collect, for the most part, the same type of data that Performance Monitor collected. It requires less overhead and is more convenient because it is GUI-managed from Management Central of the Operations Navigator interfaces.

In this re-engineering process, the QAPMJOBS file was split in two parts, including QAPMJOBMI for System Licensed Internal Code (SLIC)-oriented staff to report data for tasks and threads, and QAPMJOBOS for XPF staff to report data on a job level. QAPMJOBL is a logical file built on top of QAPMJOBMI and QAPMJOBOS and provides a view that is compatible with QAPMJOBS. The CPU time is distributed among different functional SLIC tasks. Some examples of these tasks include storage management page-out tasks, asynchronous disk I/O tasks, and communications and workstation tasks. On many AS/400 systems, SLIC tasks may use between 6% and 12% of the used CPU time, while OS/400 subsystem and monitor jobs normally use between 1% and 3% of the used CPU time.

To measure CPU utilization, we create a local area network (LAN) island that only consists of one AS/400 server as the HTTP server with a Token-Ring card installed, and two clients running Windows 95 and Windows NT. Both clients are installed with a Web client workload simulator to simulate the number of users and user activities.

Some tools can record the sequence activities that the expected user will do, including delay time from one user to another. With an isolated and dedicated LAN, we assume that there is no latency and delay time between the server and client, or latency caused by a router, DNS, or quality of the communication line.

While taking the measurement, we ran Performance Monitor tools. To learn more about setting up and running up these tools, refer to *Performance Tools for AS/400*, SC41-5340.

When you submit the Start Performance Monitor (STRPFRMON) command, you can assume that it will run for two hours of testing. The time interval for data collection is five minutes. You can set your options from the STRPFRMON screen as shown in Figure 20.

```
                    Start Performance Monitor (STRPFRMON)

Type choices, press Enter.

Member . . . . . .   *GEN          Name, *GEN
Library  . . . . . > HTTPPERF      Name
Text 'description' > 'HTTP Performance Monitor Data'

Time interval (in minutes) . . . > 5            5, 10, 15, 20, 25, 30, 35...
Stops data collection  . . . . .   *ELAPSED      *ELAPSED, *TIME, *NOMAX
Days from current day  . . . . .   0            0-9
Hour . . . . . .   2            0-999
Minutes  . . . . . .   0            0-99
Data type  . . . . . .   *ALL          *ALL, *SYS
Select jobs  . . . . . .   *ALL          *ALL, *ACTIVE
Trace type . . . . . .   *ALL          *NONE, *ALL
Dump the trace . . . . . .   *YES          *YES, *NO
Job trace interval . . . . . .   .5           .5 - 9.9 seconds
Job types  . . . . . .   *DFT          *NONE, *DFT, *ASJ, *BCH...
              + for more values
                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display       F24=More keys
```

Figure 20. STRPFRMON screen

It is better to enter words in the Text description option to distinguish the data collection. You can also specify the member instead of entering *GEN.

After you submit the STRPFRMON command, the Web load simulator terminates, and time has elapsed, you can see the report of CPU utilization by printing the System Report. Submit the `GO PERFORM` command. Then, enter option `3` (Print Performance Report). The screen shown in Figure 21 appears.

```
                          Print Performance Report

Library . . . . . .   HTTPPERF

Type option, press Enter.
  1=System report   2=Component report   3=Transaction report   4=Lock report
  5=Job report   6=Pool report   7=Resource report   8=Batch job trace report

Option   Member       Text                                  Date       Time
  1      Q992431159   HTTP_PERF_WITHPROXYCACHE_600CLNTS      08/31/99   11:59:06
         Q992431119   HTTP_PERF_WITHPROXYCACHE               08/31/99   11:19:29
         Q992431016   HTTP_PROXY_NOCACHE                    08/31/99   10:16:30
         Q992421610                                         08/30/99   16:10:37
         Q992421214                                         08/30/99   12:14:51
         Q992391553                                         08/27/99   15:53:32
         Q992241632                                         08/12/99   16:32:34
         Q992241615                                         08/12/99   16:15:25
         Q067134627                                         03/08/99   13:46:27
         Q020100002                                         01/20/99   10:00:02

                                                                       More...
F3=Exit   F5=Refresh   F11=Work with your spooled output files   F12=Cancel
F15=Sort by member     F16=Sort by text
```

Figure 21. Print Performance Report screen

After you select a member, press F6 two times. A message appears at the bottom of the screen indicating that your job was already submitted. To see the report, you can use the Work with All Spool File (WRKSPLF) command. Then, enter option 5 (display) next to your spooled file that was just created. It will look like the example shown here:

```
 Interactive Workload
     Job                Number
     Type            Transactions
 -------------     ----------------
 Interactive                     8
 PassThru                      114
 Total                         122
```

You may notice that negligible or no interactive transactions were processed. In the Component Report, you can see that the job type that mostly occurred was a batch job. QTMHHTTP user ran under the batch job type.

```
A Non-Interactive Workload
    Job         Number
    Type        Of Jobs
 ----------    ---------
 Batch              109
 Spool                4
 AutoStart            7
 Total              120
```

TCP/IP jobs are created from job descriptions and associated classes. The job descriptions and classes should be adequate in most cases. However, they may be changed to fit your configurations. The TCP/IP job description, classes, and subsystem descriptions can be found in the QTCP or QSYS library.

CPU utilization does not exactly represent the performance of Web server. Please refer to the previous paragraph. However, performance reports, combined with tools installed on the client side (for example Net.Medic from VitalSigns Software, to sense the page response time), or creating some JavaScript, allows you to measure the Web page rate from server to client.

Assuming that the network contributes to 0% latency, you can sense the correlation between CPU Utilization and page response time (the time between submission from the client to appear in the client browser).

Remember, CPU is not the only factor of performance. You should also consider communication IOP.

3.1.2 Communication IOP performance

The main LAN performance indicators that are analyzed using the information available in the OS/400 Performance Monitor database files are shown here:

- Line utilization
- IOP utilization

These factors may be related to the total system Web server performance. In some cases, IOP performance may be the most important factor for the entire Web server performance. CPU performance is less important when the load is well below the maximum utilization capacity. In slow network bandwidth,

increasing CPU performance (for example, doubling the processor) can hurt Web server performance.

A TCP/IP job is another job that is located in the QTCP or the QSYS library. As a job in Performance Tools, it creates a database to record activities. The QAPMJOBS file records batch use for jobs that are not related to communications. As a result, the batch use of a line or TCP use does not appear in the QAPMJOBS file, but only shows transaction jobs. The communications line connected to the job is classified as interactive. No batch use for communications is recorded by the QAPMJOBS file.

The IOP traffic database is located in:

- **QAPMCIOP**: Communications Controller IOP Data File
- **QAPECL**: Token-Ring LAN Statistic File
- **QAPMSTNL**: Token-Ring Station Entries File
- **QAPMETH**: Ethernet LAN Statistic File
- **QAPMSTNE:** Ethernet Station Entries File

For more details, you can make your own query (see Appendix B of *AS/400 Communication Performance Investigation - V3R6/V3R7,* SG24-4895) from those files to separate and capture TCP/IP traffic from another network protocol traffic. If your LAN design only consists of a single network protocol, for example TCP/IP, Performance Tools can represent the performance.

3.1.2.1 Line utilization

Line utilization measures the percentage of elapsed time during which the LAN was busy transferring data.

During peak hours of servicing HTTP, it is possible that LAN utilization performs high and results in poor response time and throughput due to excessive queuing. Line utilization can be displayed when you use the Display Performance Data (`DSPPFRDTA`) command after collecting performance data with OS/400 Performance Monitor. It measures the percentage of elapsed time during which the LAN was busy transferring data.

The DSPPFRDTA command provides a combination of system, component, and resource report information. Press PF21 on the Display Performance Data display to view the Communications Line Detail display.

You can issue a command as explained here:

1. Type `DSPPFRDTA`.
2. Select the performance member.
3. Select the time interval.
4. Press PF21 to display the communication line detail.
5. Enter `7` (Display Communication Interval Data) next to the line you want to view.

The line utilization for the sample interval selected is the value listed under the Percentage busy column heading.

3.1.2.2 IOP utilization

IOP utilization is the percentage of elapsed time during which the IOP was utilized. This is an important performance indicator for keeping utilization below the threshold.

To view the IOP utilization, you can print from either the Component Report or the Resource Report for performance data.

The component report has an IOP utilization report over the total report period. To relate the IOP information provided by the Component Report to particular communications line descriptions, you can use the System Report at the Communications Summary heading. By selecting the line description you want, the System Report provides the IOP resource name, IOP type, and line name.

Alternatively, you can enter the Work with Hardware Resources (WRKHDWRSC) command, with option *CMN, to display the addresses of the IOPs.

To correlate with Web traffic performance and to reduce the complexity of traffic inside the IOP card, it is easier to analyze whether the IOP only allows TCP traffic. Performance data records all traffic through the IOP, regardless of the type of protocol. SNA traffic is the easiest to capture by Performance Tools Data, because it is easy for the Performance Tool to separate SNA traffic from other traffic. Refer to Figure 22 for an example of an SSAP listing. SNA traffic uses SSAP number 4. TCP traffic is categorized as *NONSNA, which is the same case for NETBIOS and IPX, and so on.

```
----Source Service Access Points-----
   SSAP      Maximum Frame    Type
    04         *MAXFRAME      *SNA
    12         *MAXFRAME      *NONSNA
    AA         *MAXFRAME      *NONSNA
    C8         *MAXFRAME      *HPR
```

Figure 22. SSAP number for SNA and NONSNA

3.1.2.3 LAN congestion

LAN congestion affects LAN throughput degradation. With the same screen for LAN utilization, in the Congestion column, there are three different sub-columns. For more details about the congestion that was generated, you can see the report under headings listed here:

- **Local Not Ready:** The Receiver Not Ready frame transmitted by the host as a percentage of the Information Frames received by the host.
- **Local Sequence Error:** The Reject frames transmitted by the LAN IOP as a percentage of the Information Frames received by the AS/400 system.
- **Remote Not Ready:** The Receiver Not Ready frames received by the AS/400 system as a percentage of the Information Frames transmitted by the LAN IOP.
- **Remote Sequence Error:** The Reject frames received by the host as a percentage of the Information Frames transmitted by the host.

High *local congestion values* indicate that the AS/400 IOP or system data buffers are not emptied fast enough to cope with the received traffic. High *remote congestion values* indicate the need for further investigation to determine which remote stations are experiencing the congestion.

When overflow receives the buffer, it signals that the host is using a Receive Not Ready frame. This usually indicates a slow down or temporary unavailability to

receive additional frames. If a receiving station finds an error in the frames, it signals the sending station using a Reject frame for retransmission. This is an indication that the information frames are received out of sequence.

3.1.2.4 Medium Access Control (MAC) errors

The number of MAC errors is an important indication of a frame transmission, reception errors, and frame format recognition problems, such as frame validity-related errors. This data is most useful for LAN administrators as opposed to Web administrators.

3.1.2.5 Retransmission

When receive congestion errors occur on any server and information frames are not able to be processed, retransmission of those frames is necessary. When significant numbers of frames are retransmitted, it is important for you to determine the rate of transmission and compare the total number of frames retransmitted per total elapsed time.

3.1.2.6 Timeouts

If your server TCP timeouts are shorter than client TCP timeouts, the server tries to retransmit the packet to the destination until it is successfully transmitted. This generates other traffic through the LAN, which generates unnecessary congestion on the LAN.

Performance Tools/400 provides some tools that can help you to retrieve response timeouts, retry count information, and MAC errors. You can obtain information by following these steps:

1. Type `DSPPFRDTA`.
2. Press PF21 to display the communication line detail.
3. Enter `7` (Display Communication Interval Data) next to the line that you want to view.
4. Press PF11 for View 2.

You can also print this information using the Print Resource Interval Report (`PRTSCRPT`) command.

3.1.2.7 Ethernet consideration

Certain configurations of the 2838 – 10/100 Mbps Ethernet cards allow you to run the IOP with TCP/IP only, instead of all protocols, for better performance. You need a 2838 Ethernet card with either:

- 2810 IOP
- 2809 IOP

The 2838 must be the only input/output adapter (IOA) on the IOP. If you have one of these configurations, you can use the TCPONLY parameter when you create or change your Ethernet line descriptions. Setting TCPONLY to `*YES` in other hardware configurations has no effect on the line.

3.1.3 Main memory and disk arm

Memory wait time is a key component of performance, but it cannot be directly measured with any of the performance tools. However, the effect of memory

demand can be observed, measured, and controlled to a certain degree using the storage pool page fault rates.

You can observe main memory activity interactively by using the Work with System Status (WRKSYSSTS) command, or by analyzing the System Report using Performance Tools.

In general, the TCP/IP protocol and application code always runs in the *BASE pool on the AS/400 system. If the *BASE pool is not given enough storage, TCP/IP performance can be adversely affected.

On a system that performs well, the sum of database and non-database (NDB) page faults in each storage pool is usually less than 50 to 200 faults per second, depending on the AS/400 model. A higher performing system generally generates more page faults per second.

One rule of thumb for you to remember is that insufficient main memory shared by jobs can cause increased CPU and disk usage. It can lead to diminished throughput and response time.

3.2 Network data

The performance parameters include:

- Latency (the time between making a request when you see the result)
- Throughput (the number of items per unit time)
- Utilization
- Efficiency

Each of these is further explained in the following sections.

3.2.1 Network latency

The two key elements of network performance are latency and throughput. Latency is measured by Round-trip Time (RTT), which measures how long it takes for a packet to be sent from the client plus the time a response from the server takes to be received by the client. It is independent of the object size.

Response time measures how long it takes the network to service a traffic request. Network conditions can reduce the Web server's perceived performance. Significant delays occur because the HTTP protocol requires many small-packet round trips, particularly when the network path is congested.

One aspect of network latency is router performance. Routers can accept an incoming packet into a buffer, look at the header, and decide where the packet should be routed. They can also wait for an open slot before the packet is sent out. One way you can improve performance is to make the Web server with a strong packet and allow a minimum hop from the router.

The simple way to measure network Round-trip Time (RTT) is to use the *ping* command. Using the ping command is not totally representative of HTTP performance, because ping uses Internet Control Message Protocol (ICMP) rather than using TCP protocol, which is used by HTTP protocol.

A router ignores an ICMP packet and sends a TCP packet. Ping is not accurate for measuring a latency HTTP server network. Telnet can be used for measuring the feeling of latency. Telnet echoes back what you type to the server. The time between when you type and when it is echoed back.

Typical latency versus throughput is flat up to the certain value. The certain value in most cases is close to the maximum throughput of the network card devices. The point close to the maximum throughput will also reach an infinite number.

Figure 23 shows a typical increasing throughput diagram.

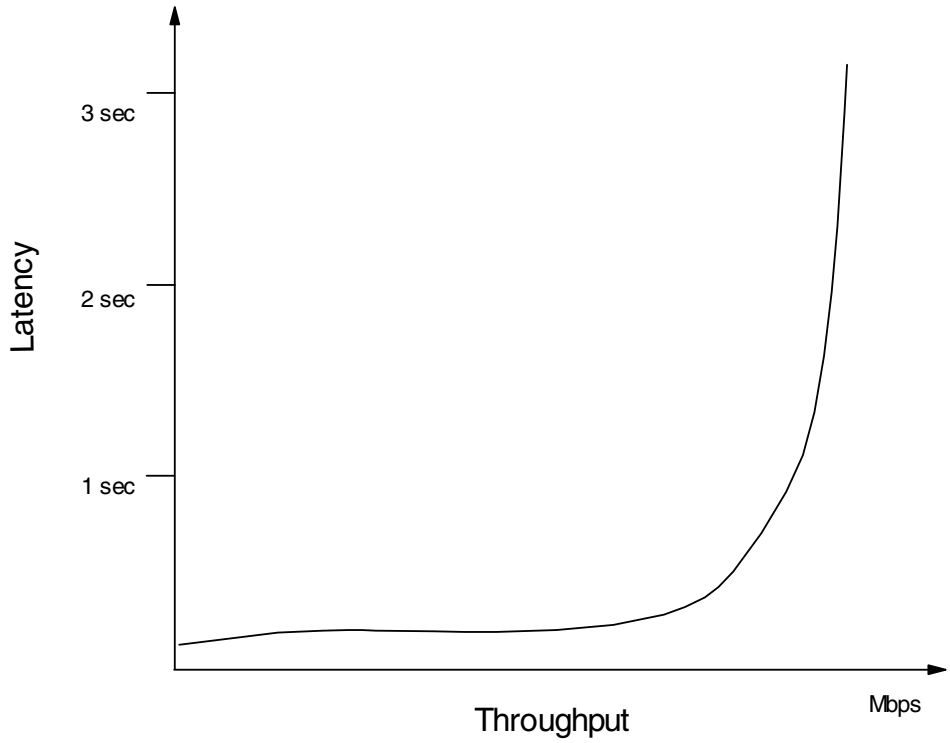


Figure 23. Typical latency versus throughput

3.2.2 Throughput

Throughput measures the time it takes to send data, up to the carrying capacity of the data pipe. Improving throughput is simply a matter of employing faster and higher bandwidth networks. Throughput also measures the maximum amount of data that a server can send through all open connections. During a given amount of time, the total number of bytes transferred per unit time is recorded.

The perceived throughput of a server can be limited by the network between the server and the client, including such factors as the bandwidth of network N gateways, routers, and network congestion. Bandwidth is limited by the bottleneck of the carrying capacity of the least capacities link or node of the data pipe.

3.2.3 Utilization

Utilization is the calculation of the number of bytes seen over a one second interval. This value is adjusted by adding size information to the appropriate

Medium Access Control (MAC) header and footer data. From this point, the amount of data is compared to the maximum theoretical throughput of your Network Interface Card (NIC) driver reports (for example, 4 Mb, 16 Mb, 10 Mb, 100 Mb, or what your topology reports).

LAN utilization in a switched environment is different than in a non-switched environment. The bandwidth of any switch depends on the mixture of port speeds, the number of ports in use, and how efficiently the many-to-many relationship is constructed.

For example, the maximum theoretical bandwidth of a switch with eight 10 Mb ports would be 50 Mb. The maximum theoretical bandwidth of a switch with eight 10 Mb ports and two 100 Mb depends on which ports are talking to which other ports. In the best of all cases (where two 100 Mb ports are talking to each other), the maximum theoretical switch bandwidth (or throughput) would be 150 Mb. If for some percentage of the time the systems on the 100 Mb ports are speaking to 10 Mb devices, the maximum throughput is down to 60 Mb.

As you can see, the idea of bandwidth utilization in a switched environment becomes one of throughput and how close to a maximum theoretical throughput the switch can achieve. This throughput is a good indicator of how efficiently your switch is using its resources. The actual number ranges quite a bit, depending on which ports are talking to which other ports.

3.2.4 Efficiency

Throughput is divided by utilization. We must measure real throughput by using the total amount of Transmission Control Protocol (TCP) transport. To sense the efficiency, you can run File Transfer Protocol (FTP) on the server as shown in Figure 24.

```
ftp> open asm20
Connected to asm20.
220-QTCP at as20.itsoroch.ibm.com.
220 Connection will close if idle more than 5 minutes.
User (rchasm20:(none)): pgreg
331 Enter password.
Password:
230 PGREG logged on.
ftp> ascii
200 Representation type is ASCII nonprint.
ftp> cd /home/gwhttp
250 "/home/gwhttp" is current directory.
ftp> get gwaccess.Q0990820
200 PORT subcommand request successful.
150 Retrieving file /home/gwhttp/gwaccess.Q0990820
50 File transfer completed successfully.
2861630 bytes received in 3.33 seconds (860.64 Kbytes/sec)
```

Figure 24. Using FTP to sense the network

Based on the information in the last row (Figure 24), note the following information:

```
8 x 2,861,630 bytes = 22,893,040 bit
22,893,040 / 3.33 = 6,874,786.79 bit per sec. or about 6.8 Mbps
```

For example, both the client and AS/400 server use a Token-Ring card and set the ring speed to 16 Mbps. Utilization is at 100% during network bandwidth. This happens in the daytime, so you should remember a lot of traffic and various LAN topology in there.

From this data, with the assumption that Token-Ring efficiency is about 60% to 80%, the efficiency is:

```
( 6.8 / (0.80 * 16)) * 100% =  53%
```

In this example, we show the simplest way to sense network efficiencies. In 3.2.5, “Network measurement tools” on page 37, we discuss network measurement.

3.2.5 Network measurement tools

To help you sense or measure the network performance of the system on which you are working, we give you two examples of tools that are available in the marketplace network analyzer. These tools measure the network environment where your AS/400 server is located.

3.2.5.1 Distributed Observer

Distributed Observer is a windows-based network analyzer from Network Instrument, LLC. We use this tool to observe:

- Network and bandwidth utilizations
- Network trend
- TCP utilization and IP subprotocols applications
- Internet activity

There are many observation options in this software, but they depend on your intention to use them. We only selected some of the options that can guide us to our goals.

Before measuring the traffic by using Performance Tools/400, it is easier to separate IP traffic from other traffics. To prove and observe that only Internet traffic happen during performance measurement, you can use an IP subprotocols monitor as shown in Figure 25 on page 38.

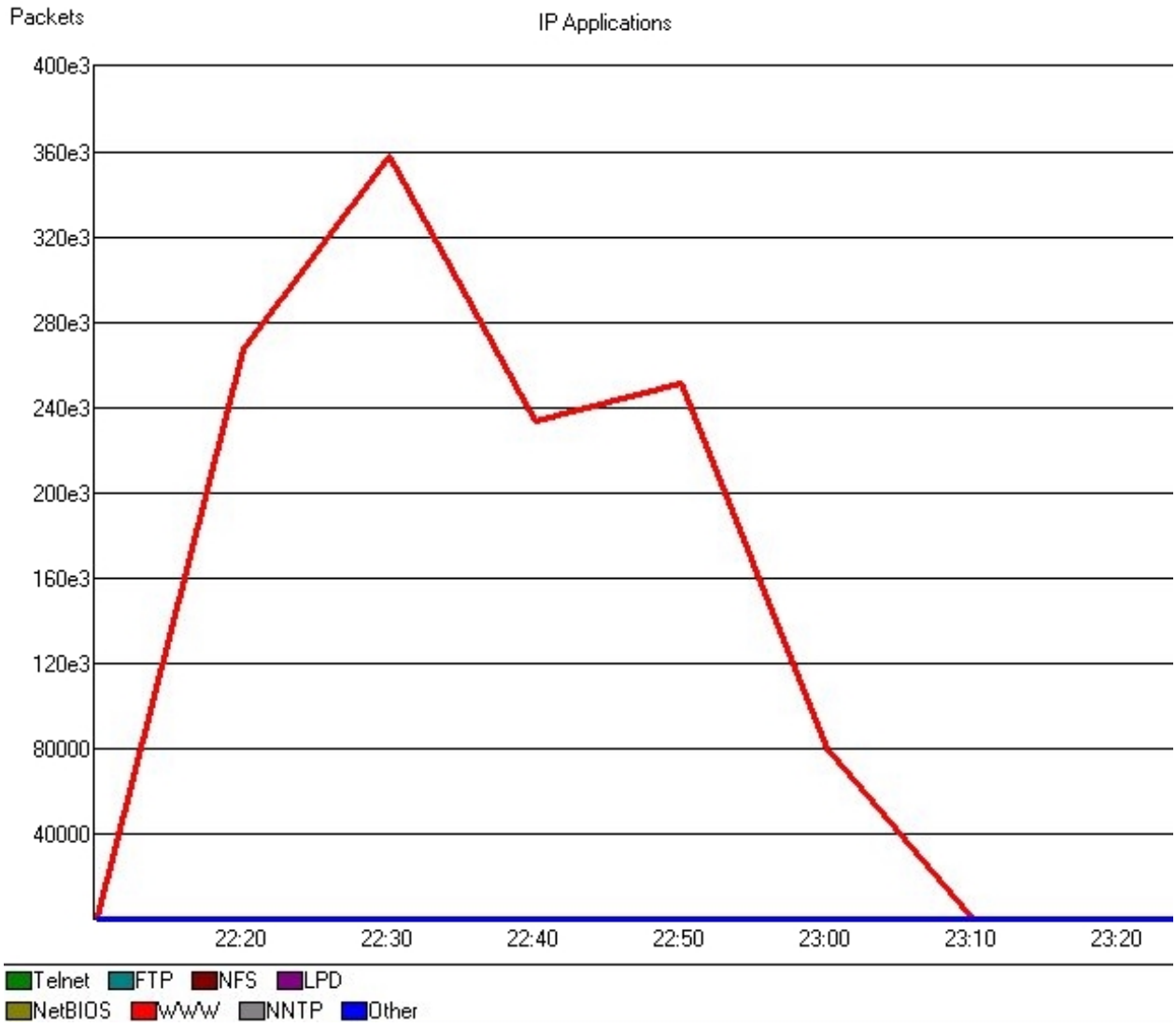


Figure 25. IP applications graphic monitor

As shown in Figure 25, you can see that no other protocols are running in this observation. Even for IP traffic, only Internet traffic occurred. There was no Telnet, FTP, LPD, NNTP, or NetBIOS over TCP/IP. The Y-axis represents the amount of the packet that was transferred per second. You can relate this data to the number of bytes that were transferred during the observation. To calculate the number of bytes transferred from Internet subprotocols, see the Web analyzer tools in 3.5.2, “AS/400 Web analysis tools” on page 49.

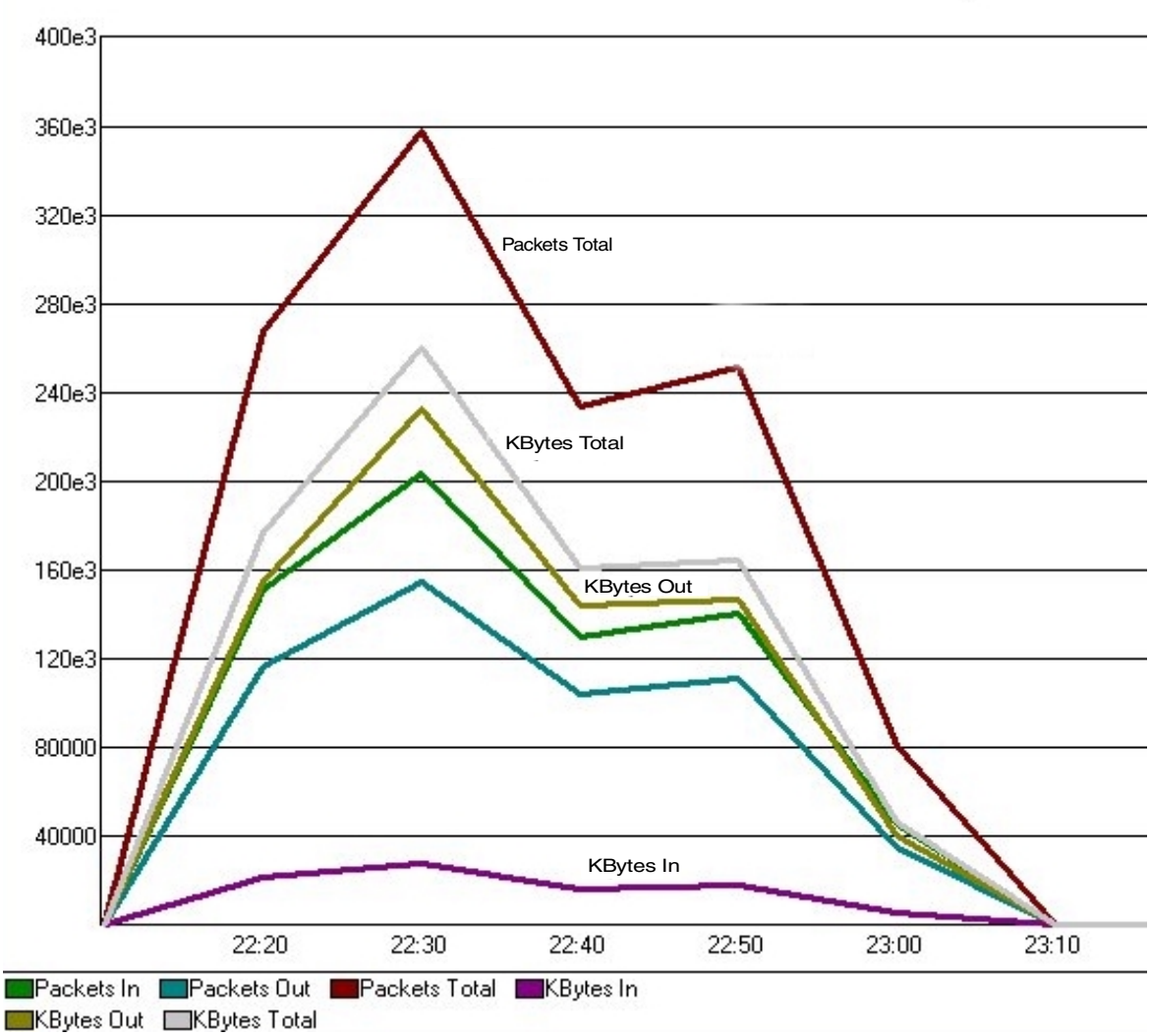


Figure 26. Packets and KBytes traffic

Figure 26 shows the correlation between Internet packet traffic and KBytes Total. The traffic was captured from the AS/400 IP address, instead of the entire LAN traffic. Figure 25 on page 38 shows you the Internet packet traffic. By looking at Figure 26, you can see the correlation from the packet traffic to KBytes Out and KBytes In through the Web server.

The example in Figure 27 on page 40 shows how to determine the number.

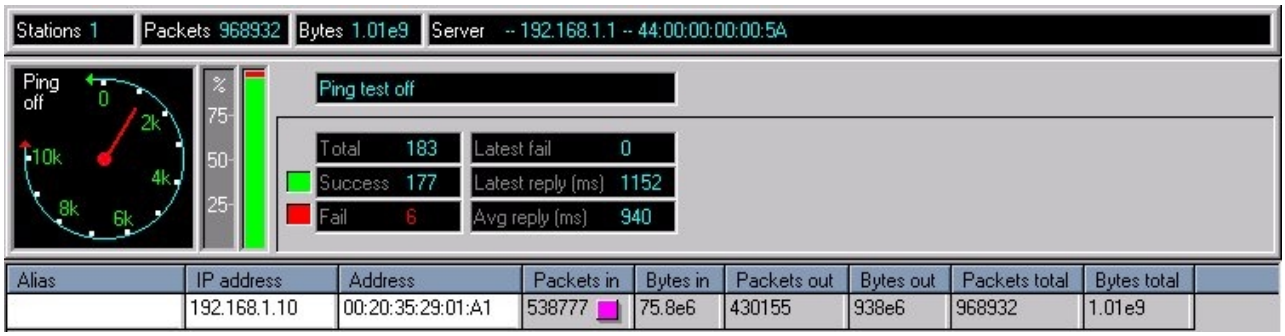


Figure 27. Total numbers of traffic

This KBytes Out should have the same amount of data that we get from accessing log files when they are analyzed by using Web log analyzer software. This does not separate the kind of method used for generating traffic.

In some cases, you may want to know when the maximum traffic occurred. To determine this, refer to the graph shown in Figure 28.

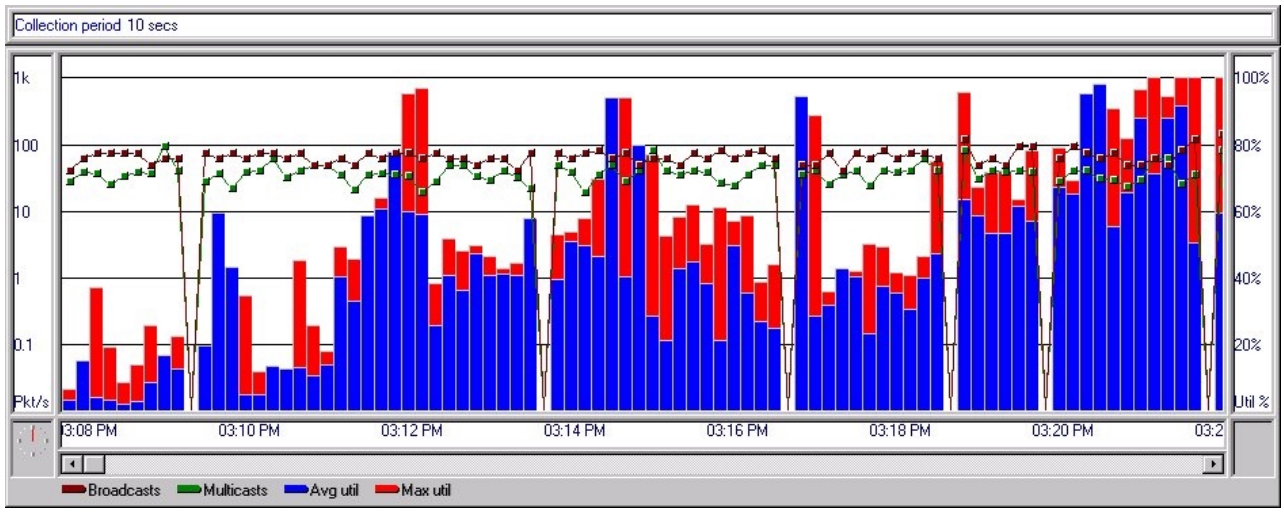


Figure 28. Network activities graphics

Each bar represents a 10 second average of traffic. You can see when the maximum traffic occurred. Maybe this traffic was generated by Web activity. If you are sure that the traffic was generated by a Web server, you can see what really happened in the AS/400 IOP or LAN card. With the same time frame, for example at 03:16:40 hours, the traffic was at a maximum. Using the same time frame at the Performance Report on Trace Component section report under Transition Options (*TRSIT), you can see what our Web server activities are. You can obtain your answers by asking these questions:

- Do my Web activities generate a lot of network traffic?
- What kind of jobs generate the traffic?
- Is my LAN IOP adequate enough when maximum network activities appear?

3.2.5.2 NetBoy

NetBoy comes from NDG Software, Inc. It is part of the software called WebBoy, which is used to capture the Web traffic. Web traffic captures and makes a

graphic of URL Access and Network Load. WebBoy provides statistics on standard Web traffic, including the URLs accessed, cache hit ratios, Internet protocols, and user-defined protocols. This tool can be used to measure Web traffic, byte transfer, and sub-TCP protocol traffic.

Another tool in this software packet is called PacketBoy. This tool is used to capture the packet traffic and network traffic. PacketBoy is a packet analyzer and decoder package that is capable of decoding many commonly used LAN protocols.

3.3 Client data

One user expectation is the waiting time between submission and the time the data is displayed in the browser. The shortest waiting time is equal to the time when the user opens the Web pages from their local disks. A client's resource usage depends most on the usage of the client browser. For some operating systems, we need another tool to measure the CPU utilization and memory usage. Some operating systems offer it as part of the operating system. Better CPU speed and big enough memory generally produce better client performance.

Unfortunately, the increase of CPU speed and the maximum amount of memory available do not indicate an increase in networking speed or throughput. The greatest bottlenecks occur on network devices. A faster network card does not guarantee faster response time. It depends on the effectiveness of data transmitted through the network card.

To measure the impact of network performance, you can issue the `ping` command to your destination server.

Another aspect of user expectations is the perception of performance. The perception of performance depends on the client side, including the client platform and operating environment, and the Web client. Some Web clients take steps to improve the perception of performance. For example, inline images are displayed while they are downloaded, instead of the client waiting until the entire image is downloaded.

To measure the real response time of the Web server, you should not depend on the Web browser characteristics. Use Telnet to HTTP port (80 is a common port for HTTP), and then submit the `GET` command to get the index or welcome file with HTTP attribute, for example:

```
telnet www.myserver.com 80
GET /home/index.html HTTP/1.0
```

The HTTP attribute depends on the HTTP standard you implemented, either HTTP/1.0 or HTTP/1.1.

Another PC-based tool to measure the total time served from the time a request was made to displaying it on the browser is discussed in 7.5.1, "General network considerations and intranet considerations" on page 147.

3.4 Web server access log analysis

Common Web servers are always equipped with log files, such as access logs, client logs, referrer logs, and error logs. These logs record and track transactions, traffic, time, methods, the user, the origin of user, and so on. The log generates automatically by the end of the day. From this log, for example, we can calculate the number of hits per day, users per day, bytes transferred per day, pages accessed per day, and so on. You can also resume the log, for example, for one week or one month.

These log files do not directly represent AS/400 Web server performance. As described in 1.5.1, “Measurement” on page 14, by using both log files and Performance Tools, you can determine the performance of the Web server.

3.4.1 Common access log file format

OS/400 V4R4 implements common access codes, data description specifications (DDS), and the extended access code format. The common log format keeps three separate log files that are not always processed correctly by all analysis tools.

The Web server generates log files that document all activity on your site, such as:

- The IP address or domain name of the visitor to your site
- The date and time of their visit
- The pages they viewed
- Any errors that were encountered
- Any files downloaded and the sizes
- The URL of the site that referred to yours, if any
- The Web browser and platform (operating system) that was used

The common log file format is:

```
remotehost rfc931 authuser [date] "request" return-code bytes
```

Remotehost, or the client origin based on the IP address or domain name, combine with the date (time included) and the bytes transferred. For example, you can make a simple calculation of how fast the network bandwidth is from the server to the client.

An example of the contents of a log file is shown in Figure 29.

```
9.5.99.118 - - [03/Nov/1998:09:07:58 +0000] "GET /home/george/services.html HTTP/1.0" 200 232
9.14.3.71   - - [03/Nov/1998:09:19:06 +0000] "GET /QIBM/NetworkStation/nsmgr.htm HTTP/1.0" 200 180956
9.5.99.118 - - [03/Nov/1998:10:07:42 +0000] "GET /home/george HTTP/1.0" 403 238
9.5.99.118 - - [03/Nov/1998:10:07:42 +0000] "GET /home/miscweb/as400.gif HTTP/1.0" 401 186
9.5.99.118 - - [03/Nov/1998:10:58:14 +0000] "GET /home/george/products.html HTTP/1.0" 304 0
9.5.99.118  - - [03/Nov/1998:14:48:22 +0000] "GET http://beanmachine.raleigh.ibm.com/ HTTP/1.0" 403 237
```

Figure 29. Common log format

The logged fields include:

- **Remote host/user address**: IP address or domain name of the user who accesses it.
- **RFC931**: The remote log name of the user. It is usually blank.
- **User authentication**: The user name for the user if it is required for access to the site.
- **Date/time**: When the user accesses the server content, the default time offset is GMT.
- **Method requested**: The Method token indicates the method to be performed on the resource identified by the Request-Universal Resource Locator (URL).

```
Method     "GET"
           HEAD"
           "POST"
           extension-method (token)
```

- **Return code**: Indicates the event that happened. The first digit of the status code defines the class of the response. The last two digits do not have any categorization role. There are five values for the first digit:
  - **1xx: Informational** – Not supported in HTTP/1.0.
  - **2xx: Success** – The action was successfully received, understood, and accepted.
  - **3xx: Redirection** – Further action must be taken to complete the request.
  - **4xx: Client Error** – The request contains bad syntax or cannot be fulfilled.
  - **5xx: Server Error** – The server failed to fulfill an apparently valid request.

  The individual values of the numeric status codes defined for HTTP/1.0 and an example corresponding set are presented here:

```
Status-Code     "200"   ; OK
                "201"   ; Created
                "202"   ; Accepted
                "204"   ; No Content
                "301"   ; Moved Permanently
                "302"   ; Moved Temporarily
                "304"   ; Not Modified
                "400"   ; Bad Request
                "401"   ; Unauthorized
                "403"   ; Forbidden
                "404"   ; Not Found
                "500"   ; Internal Server Error
                "501"   ; Not Implemented
                "502"   ; Bad Gateway
                "503"   ; Service Unavailable
                extension-code (3DIGIT)
```

  For a more detailed description of the codes, see HTTP/1.1 specification RFC2068.

  HTTP status codes are extensible. The codes listed here are the only ones that are generally recognized. HTTP applications are not required to understand the meaning of all registered codes, although such an understanding is obviously desirable.

- **Transfer size (bytes)**: For static Web pages, this information comes from how big the file is. For dynamic, it depends on the content.

3.4.2 Extended access code format

In V4R4, the IBM HTTP Server supports the Extended Log Format as defined in the specification "Extended Log File Format W3C Working Draft".

The extended log format combines access, referrer, and agent log information into one log file. This format also allows you to configure the information that is written in each access log entry. If you do not specify a named extended log format, each access log entry will have a common format with agent and referrer information appended. Use the ExtendedLogFormat directive to specify the information that the HTTP server logs into the access log files. Using the extended log format does not affect error, referrer, or agent log files.

ExtendedLogFormat works in the IFS and in QSYS. In QSYS, since the record length is fixed, log entries are truncated if they are longer than the fixed length. A message is sent to the error log the first time this happens.

For the latest information and sample files, go to the Web site at:
`http://www.w3c.org/TR/WD-logfile`

The extended format is more detailed than the common format because it includes all of the fields of the common format and some additional fields. The extended log format is shown in Figure 30.

```
                  Work with HTTP Configuration
                                                          System:   AS25
  Configuration name . . . . . . . :   GP_01

Type options, press Enter.
  1=Add   2=Change   3=Copy   4=Remove   5=Display   13=Insert

     Sequence
Opt    Number    Entry

       00010     # * * * * * * * * * * * * * * * * * * * * * * * * * * *
       00020     ExtendedLogFormat GP_XFmt {
       00030           Software
       00040           StartDate
       00050           EndDate
       00060           Date
       00070           Field time
       00080           Field c-ip
       00090           Field cached
       00100           Field bytes
                                                                    More...
F3=Exit   F5=Refresh   F6=Print List   F12=Cancel   F17=Top   F18=Bottom
F19=Edit Sequence
```

Figure 30. Extended log format setting in a server instance

Some Web log analyzer tools cannot recognize the extended access log formats that are available in the marketplace. Be careful when deciding whether to choose the standard or extended log format after you observe the capability of your Web log analyzer.

3.4.3 Editing the logs format

This section explains how to modify the logs format. First, you can modify it directly using the Work with HTTP Configuration (WRKHTTPCFG) command and modify the directive of the server instance. Second, you can use Web-based administrator on port 2001 of your AS/400 system.

If you are familiar with using the syntax of log files, you should edit directly from the CL prompt. An easier way is to use a Web-based administrator. OS/400 directly edits your directive if you made modifications from the Web.

3.4.3.1 WRKHTTPCFG

You can specify which directive you want to apply, followed by the sub-directives for those directives. The directive using the World Wide Web Consortium Standard should be readable by many common log analyzers. You must specify the log file name of each log format and the subsequent AccessLog file. Enter:

```
WRKHTTPCFG CFG(your_http_server_instance)
```

Scroll down until you find the logging section. The syntax is:

```
ExtendedLogFormat format-name {
     Field  field-value
     subdirective
     subdirective
     subdirective
     .
     }
```

In this syntax, note the following points:

- `Field`

  This is the field subdirective. This directive requires that you specify at least one field subdirective. Use the field subdirective to specify a field to record an entry in the access log file. The HTTP server requires that you specify at least one field subdirective with the ExtendedLogFormat directive. The order of the field directives specifies the order in which the HTTP server records information in an access log entry. If the field information is not available, a dash (-) is recorded in the log entry.

- `field-value`

  One of the following elements can specify the field-value:

  – **Identifier**: Relates to the transaction as a whole.

  – **Prefix-identifier combination**: Relates to the information transfer between parties that are defined by the prefix.

  – **Prefix-header combination**: The HTTP header field for the transfer between parties.

  The following identifiers do not require a prefix:

  – `date` is the date at which the transaction completed
  – `time` is the time at which the transaction completed
  – `bytes` indicates the bytes that were transferred
  – `cached` records whether a cache match occurred; 0 indicates a miss

The following identifiers require a prefix:

- `ip` is the host IP address and port
- `dns` is the host DNS name
- `status` is the status code
- `comment` is the comment returned with status code
- `method` is the method URI
- `URI`
- `uri-stem` is the stem portion alone of URI (omitting query)
- `uri-query` is the query portion alone of URI

Specify the following prefixes:

- `c` Client
- `s` Server
- `cs` Client to server
- `sc` Server to client

For *Remark*, include and specify the comment information to be included as a directive. Use the Remark subdirective to specify that you want to include comment information in the log file header.

- `Software`

  Include the software that generated the log. On the AS/400 system, the software is IBM HTTP Server for AS/400. On other platforms that support a different type of HTTP server, this information is used to determine the type of HTTP server that generated the log.

  Use the software subdirective to specify that you want to include the name of the software that generated the log in the log file header.

- `StartDate`

  Include the date and time at which the log was started as a directive. Use the StartDate subdirective to specify that you want to include the date and time the log was started in the log file header.

3.4.3.2 Web-based administrator

To use the Web-based administrator method, complete the following steps:

1. Open your browser, and point to the URL address of your AS/400 system that you want to manage. Do not forget that a standard AS/400 system Web-based administrator uses port 2001. You will be asked to enter your name and your password.
2. On the AS/400 Tasks screen, select **IBM HTTP Server for AS/400**.
3. Click sequentially on **Configuration and Administration->Configuration.** Under the Configuration title, select the configuration with which you want to work.
4. Click on **Logging->Create extended log format**. You can either choose to create a new log file with an extended log format, empty an existing log file, or create a new log file based on an existing one.

In the window shown in Figure 31, we inserted six lines of fields. For every field value that you want to insert, click the **Apply** button at the bottom of the page. The explanations for each field, their values, and so on, are explained in 3.4.3.1, “WRKHTTPCFG” on page 45.

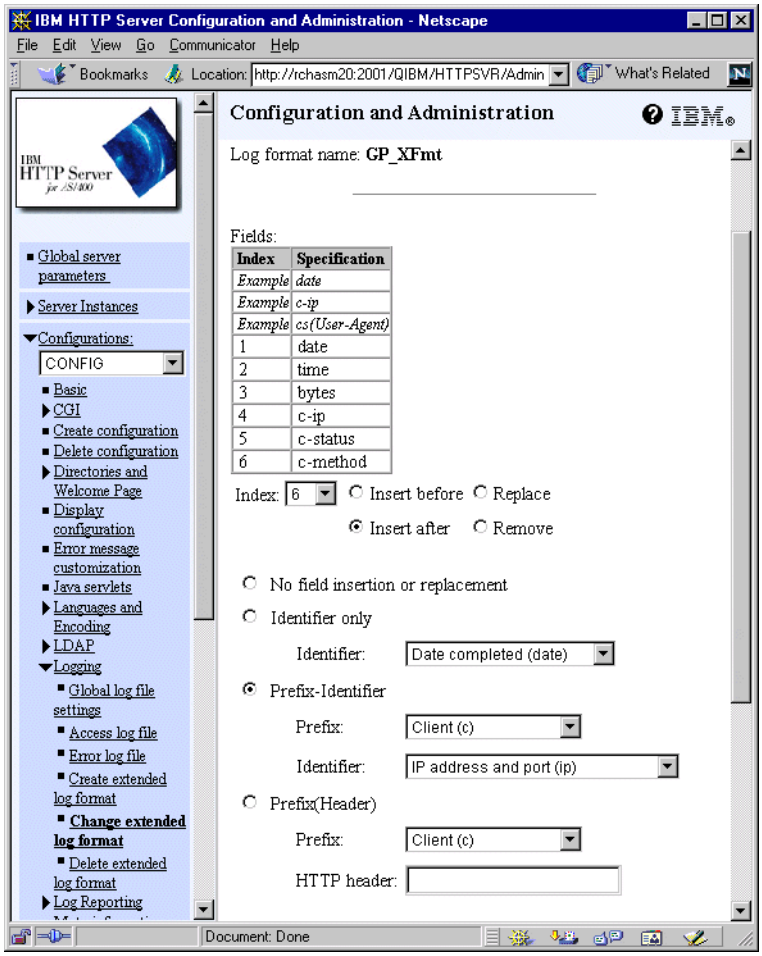


Figure 31. Extended log editor using a Web browser

The IP address, date and time, bytes, method, and cache may be useful for analyzing the AS/400 performance. Sometimes, you do not need to put the entire parameter in the extended log file. Remember, the CPU has additional tasks while serving the Web, such as creating log files. Although you have enough disk space in the AS/400 server to accommodate the log files, the Web log analyzer software takes more time to analyze large log files compared to analyzing smaller log files.

3.4.4 Other logs

The AS/400 system supports other logs, in addition to access logs, for different reasons for the Web administrator to perform analysis. The default settings of these logs use the common format. You can choose whether you want to store the agent access log file in the QSYS file system or the integrated file system (IFS). These fields are optional. The other logs are listed here:

- **Cache access log**: The server logs access requests for cached files in the cache access log file. The cache access log file contains the time, date, host name, or IP address of the client making the request. The server only writes entries to the cache access log file if the access log file is configured.
- **Proxy access logs**: The server logs an entry each time it acts as a proxy. The proxy access log file contains the time, date, host name, or IP address of the

client making the request. The server only writes entries to the proxy access log file if the access log file is configured.

- **Agent log**: The server logs the type of Web browser used by the client making the request.
- **Referrer log**: The server logs the identity of the Web page that referred to (linked to) the requested page.

3.4.5 Log maintenance

With the log maintenance options, you can specify how to handle the accumulation of daily logs for days that have past. You can choose whether you want to keep old logs, remove logs after they reach a certain age or a "collective size", or run your own program at midnight each night to handle old logs. Note that the *collective size* is the collective size of all access logs only (not combined with agent and referrer logs), all agent logs only (not combined with access and referrer logs), or referrer logs only (not combined with access and agent logs).

To reduce the space that the access, agent, and referrer logs require, you can specify that the logs be automatically removed, based on the age of the log or the collective size of the logs. If you are interested in running your own backup program to store the logs, you can specify a user exit. In this case, you specify the path to your program and the parameters to pass to your program. The server appends the path to the logs to this information.

The settings that you specify on the Access Log File Configuration form also apply to the agent and referrer logs.

3.5 Access log analysis tools

Although the AS/400 system includes a simple Web analyzer or log report, many PC tools that are currently available in the market offer a different angle for viewing and analyzing Web logs. Most of them concentrate on the user behavior and are very useful from a marketing strategy perspective.

3.5.1 What we want to get from AS/400 system log files

The idea for creating an analysis of Web logs is to answer such questions as:

- What is our Web site traffic?
- How many visitors visit our site every day?
- Is the number of visitors to our site growing?
- From what companies or countries are our visitors coming?
- Which are the most viewed or popular pages?
- How much data is being retrieved by the visitors?

The next question is: If you just want to analyze from a technical point of view, do the logs files give you enough information?

If your aim is to know more about AS/400 server performance, this log does not give us adequate information about performance. Only a small piece of information directly represents the performance, such as bandwidth utilization, the number of bytes transferred, and hits.

The only tool available on the AS/400 system to observe the performance is Performance Tools for AS/400. In fact, this tool cannot grab Web traffic data specifically. This is the reason why we should combine the use of Web log analyzer software with AS/400 Performance Tools.

In the next section, we perform sizing for predicted workloads using the same collection data.

3.5.2 AS/400 Web analysis tools

This section discusses OS/400 V4R4 log reporting and Web Trends Analyzer. Generally, Web reporting allows an administrator to know the how the HTTP server is used by clients.

3.5.2.1 OS/400 V4R4 log reporting

Since OS/400 V4R4, the AS/400 system has the ability to define a simple analysis and snap shot of Web performance and report it dynamically through an HTML file.

Each server instance that wants to use logging and reporting must have its own configuration file with its own logging and reporting directives. Reports are stored in directories unique to each server configuration file (`/QIBM/UserData/HTTPSVR/reports/instance_name`), where `instance_name` is the name of the HTTP server instance that generated the reports.

If errors occur during the generation of the reports, a file (debug.mmddyyyy) is generated in the reports directory.

Refer to *HTTP Server for AS/400 Webmaster's Guide,* GC41-5434, for more details about log reporting.

Basic reporting

The basic report is created dynamically through the use of Java applets that are accessed from a browser. Parameters of reports are easily set up using Web-based administration. To see the basic report, from the Configuration and Administrations Form pages, choose **Access Report - Basic**. Enter the configuration with which you want to work. From there, select the **Report Template** option.

The AS/400 system gives you a report on a certain starting date and time to a certain ending date and time for:

- **Host report**: To report how many times and how many bytes were accessed by the certain client.
- **URL report**: To report how many times and how many bytes that were transferred by specific URLs that were displayed.
- **Method report**: To report how many times different methods were used.
- **Code report:** To report how many times successful redirect and error codes happened.

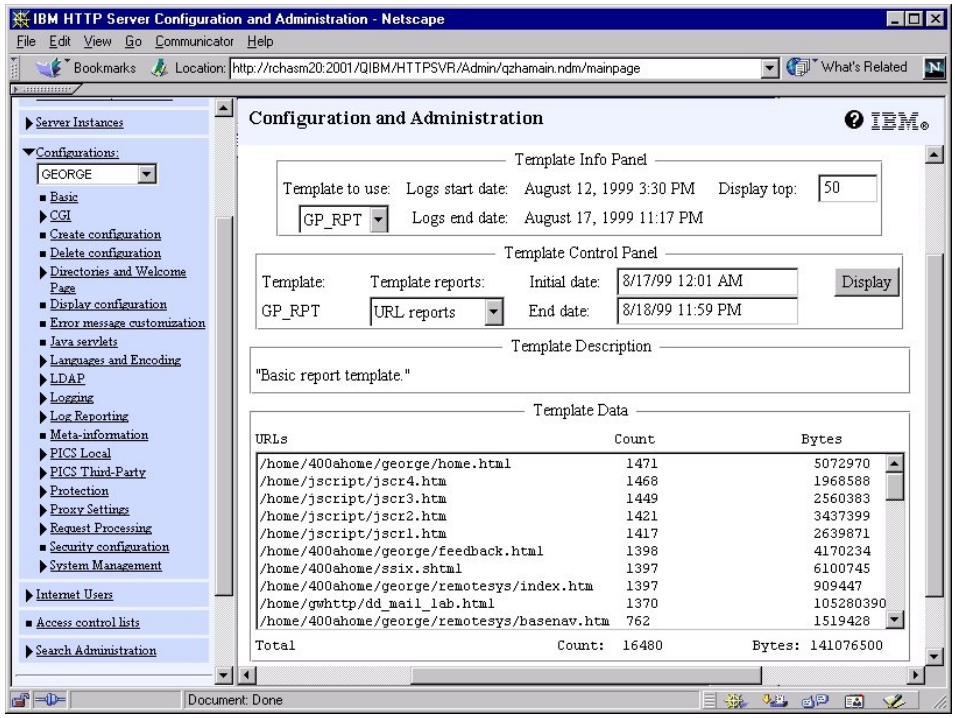


Figure 32. Basic reporting: URL report

From the report shown in Figure 32, we know the total bytes transferred out of the Web server based on the user IP address (host reports) or URL accessed (URL reports). Unfortunately, it does not give us the distribution of bytes transferred based on accessing time. You will not recognize the peak hour, peak date, and so on. If you need to get the report in a specific time frame, for example only for one hour on the peak time of the Web load, you should know when the peak hour and type occurred, including the date and time, on the Initial Date options and End Data options.

Avoid selecting a large time frame for one report, for example, a full month report.

Creating a basic report template

Follow the menus shown in Figure 33. For this template creation, we can specify which address template (for example, the IP address starting with or the domain starting with) should be included or excluded. For certain reasons, for example, to analyze customer or user behavior, you may want to exclude your intranet address. In the same way, you can include or exclude the URLs.

Another option is to select to exclude the Methods and Return Codes. By default, all Methods and Return Codes are included.

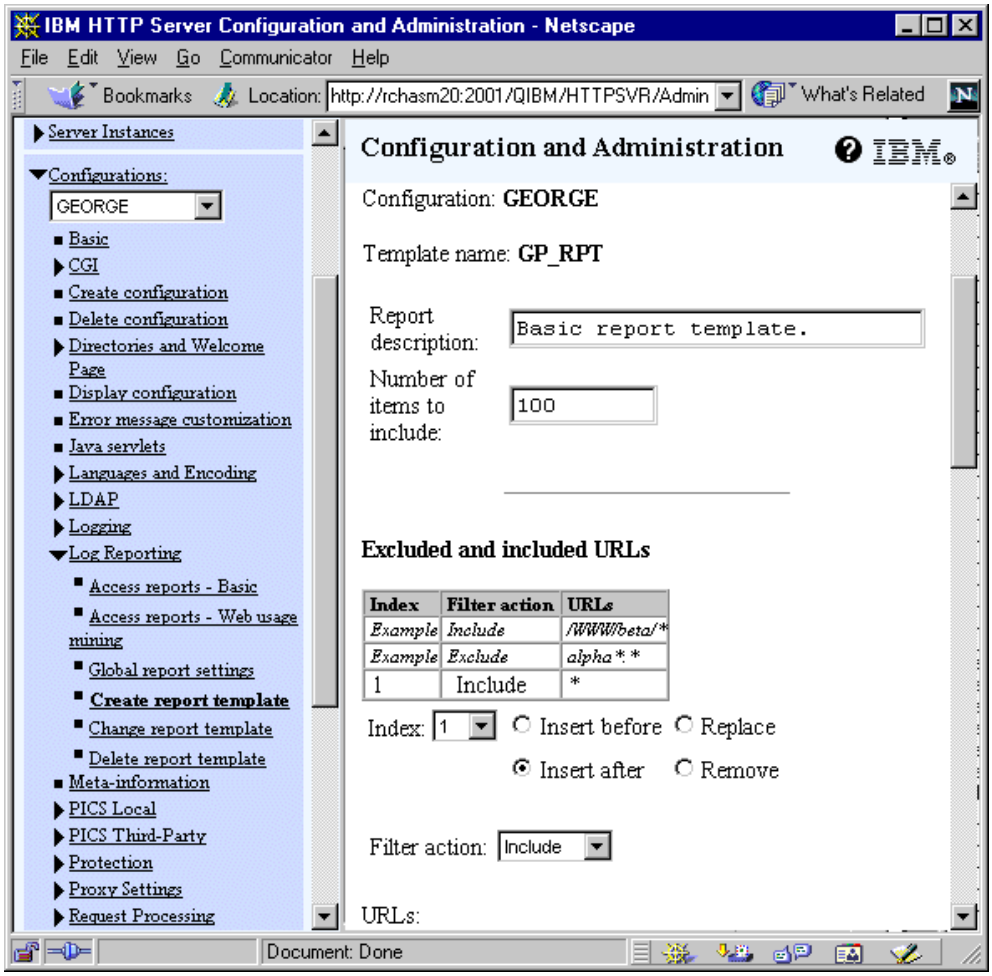


Figure 33. Create report template

Web usage mining

If you want to understand how your users navigate through your Web site, you can look at the Web usage mining statistics. They tell you the sequence of the Web pages that a user clicked through during a visit.

The reports can tell you where people enter and exit from your Web site and which Web pages, as a group, are visited most. You can see the browsing patterns and identify user behavior, which, in turn, allows you to better organize your Web pages. The reports are generated automatically and are not tailorable except through the standard report templates.

The reports calculate statistics based on user activities, paths, and groups. Since your client does not have a user profile to the AS/400 Web server, it will not be counted. We do not discuss this mining further because it does not represent AS/400 Web performance.

Web Activity Monitor

The Web Activity Monitor provides a real-time view of server activity since the last time the server was started. It is integrated with the V4R4 HTTP server.

Enabling the activity monitor is a function performed through the Systems Management options of the Administration server. It places the following directive in the server instance:

```
Service /Usage*  INTERNAL:UsageFn
```

Once monitoring is enabled, you can view the statistics through the administration server. The Work with Server Instances has a Monitor button that displays the statistics. There are four types of statistics displays initiated by the Monitor request as shown in Figure 34. These are described in the following list. Statistics counters are reset when the server is started or restarted.

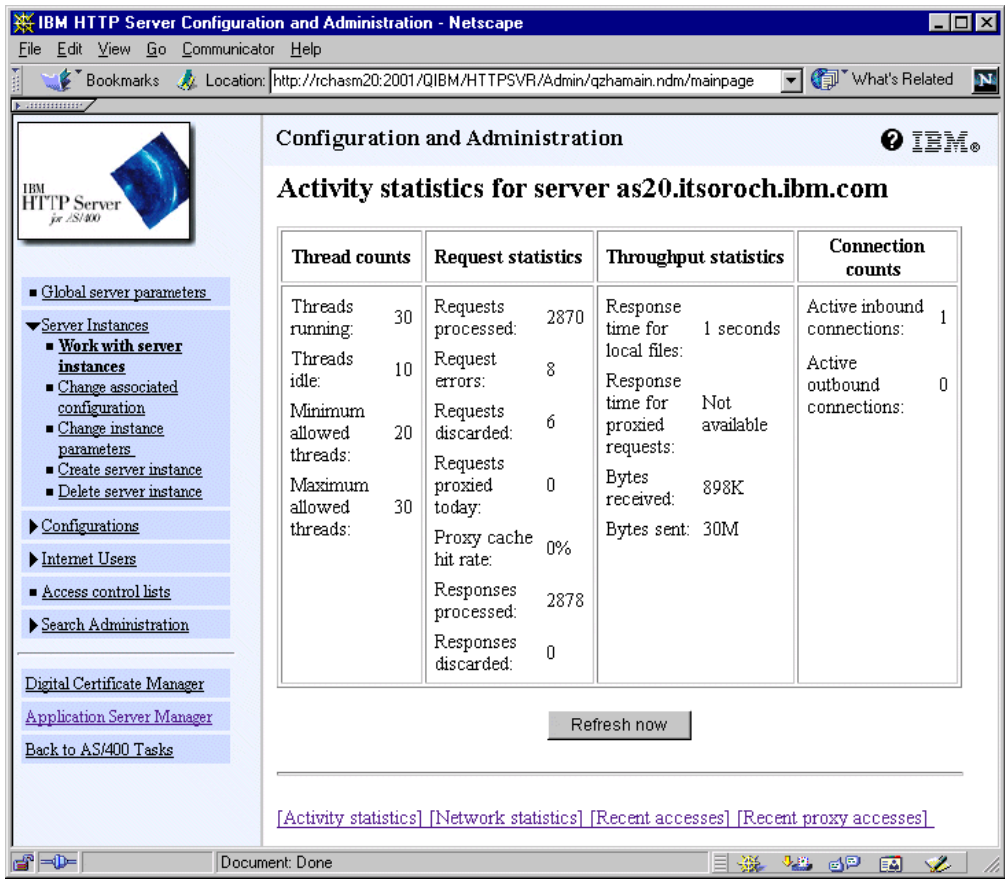


Figure 34. Web activity statistic: Activity monitor

- **Thread counts**
  - Running and idle:

    Each time your server receives a request from a client, it uses a thread to perform the requested action. If the server is performing DNS lookup, it uses two threads. When the server first starts, the number of idle threads is low and most of the configured threads are listening. As time passes, the number of idle threads increases to a point where there are sufficient threads active to sustain the workload.

  - Minimum and maximum:

    These are configured values. The number of threads impacts AS/400 system performance. Specifying too few threads can result in waiting for a

thread to become available. On the other hand, specifying too many threads can result in too many resources being allocated unnecessarily.

- **Request statistics**
  - Requests and responses processed:

    This is a basic measure of the server's activity. It also indicates the number of requests coming in and being responded to by the server.
  - Requests and responses discarded (requests are related to timeouts):

    Requests discarded indicate the number of client connections dropped due to an input timeout. This is where the client connects to the server and then fails to send a request.

    Responses discarded result from a timeout on a local file (not a CGI program) request from the server. If it does not receive the file, it drops the client connection.
  - Proxy requests and cache hit rate:

    These are measures of the server configured as a proxy.
- **Throughput statistics**
  - Response times:

    For local files, this provides a server performance measurement.
  - Bytes sent and received:

    This refers to only server requests and response data. This information can provide served data volume performance figures.
  - Unknown bytes received:

    This is the volume of data received in messages that the server was unable to identify after the connection was established.
- **Connection counts**
  - Inbound connections indicate normal incoming client connections.
  - Outbound connections are when the server is configured as a proxy.

Network statistic
Network statistics show the outgoing and incoming data traffic through the Web server. The given data is only server data. TCP/IP frame overhead is not counted. These statistics do not give detailed information like when the most traffic happens, from which station the most traffic comes, and the correlation between traffic and response time.

Recent accesses
This report comes from log access. The fields include: IP address, Date and time, Request, Response code, and Message size.

Recent proxy accesses
When the HTTP server is configured as a proxy server, this report shows proxy transactions in the same format for the Access log.

3.5.2.2 System Management setting

In this setting, you can try to adjust the performance by enabling dynamic cache, the number of threads, and the amount of time outs. Changes you make to the

Local caching form, the Performance form, and the Time-outs form all influence the performance of your server.

Performance

Figure 35 shows enabling dynamic caching, the number of threads, and persistent HTTP connection parameters.

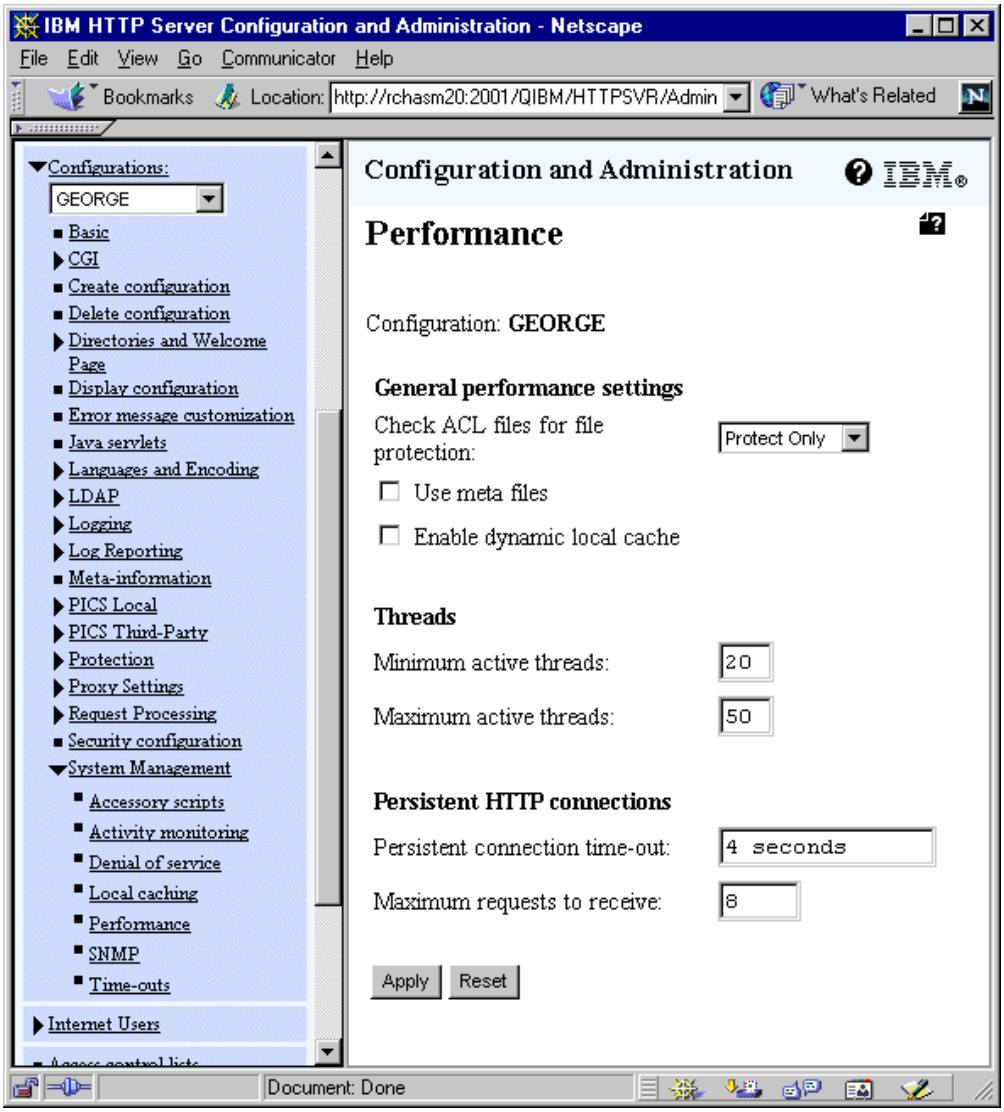


Figure 35. Web-based setting for the Performance parameters

Basically, one request occupies one thread if the server does not perform a DNS lookup. Two of the server threads perform a DNS lookup. Be careful to set the number of threads, because too many threads does not indicate good performance. Nor is it good to use only a few threads. When the request to Web server reaches the top and all of the threads are occupied, the next request is held until another request releases the threads and threads become available.

If your AS/400 system resources are efficient, for example, you have a large amount of main memory and a high CPW processor, giving the Web server a few threads is an advantage to boost performance. Continuously increasing threads

requires more memory allocation. The threads are released while the request is ended.

HTTP/1.0 does not implement a persistent connection. See 2.5, “Persistent connection” on page 23, to implement HTTP/1.0. You can set a persistent connection time-out as low as possible. HTTP/1.1 has a persistent connection. Maintaining longer persistent connection time-outs opens too many useless sockets.

For more information about threads and persistent connection time-out considerations, see 6.4.1, “AS/400 file system considerations” on page 103.

TIme-outs

For a better understanding of the Time-outs parameter, refer to Figure 36.

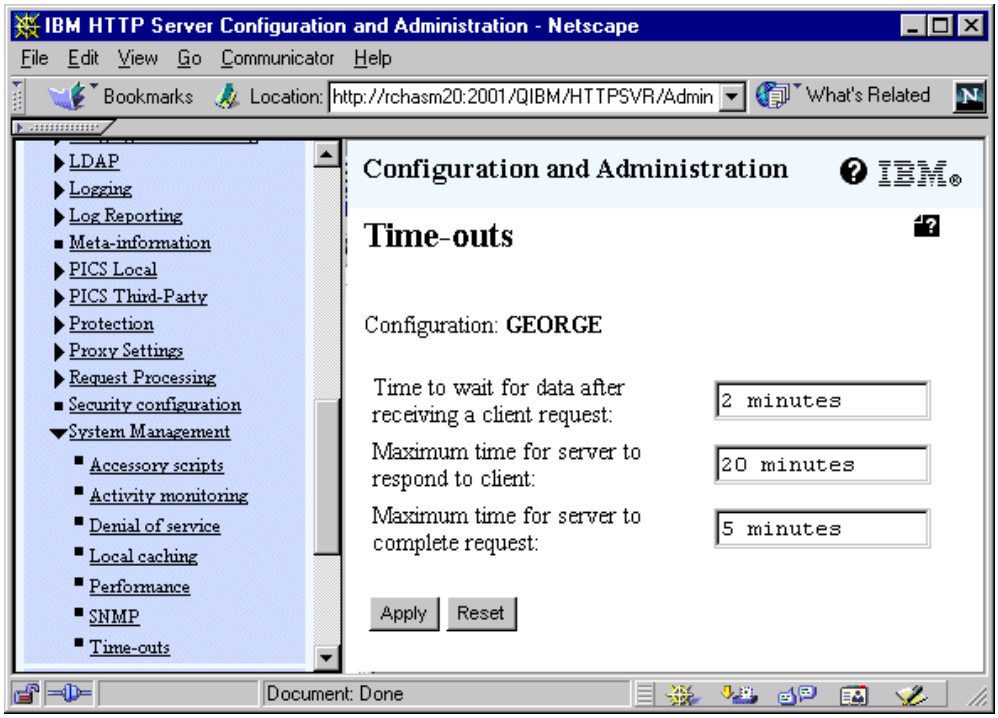


Figure 36. Web-based setting on Time-outs parameters

Time to wait data after receiving a client request indicates the time needed from the first right direction SYN arrow to first right direction DAT arrow. The different time of those transactions may not complete in two minutes, for example, the connection is dropped by the server.

Maximum time for server to respond to client represents the time of the first left direction DAT arrow to transfer from the server to the client.

To give the best number of time-outs, you should understand the characteristics of a users network. Intranet users need less time-out than Internet users.

3.5.3 PC-based analyzers

This section describes two PC-based analyzers, as an example.

The first one is the WebTrends Analyzer, from WebTrends Corporation in Portland, Oregon, which can be found at: `http://www.webtrends.com`

The second one is NetIntellect, from WebManage Technologies, Inc., which can be found at: `http://www.webmanage.com`

3.5.3.1 WebTrends Analyzer

WebTrends Analyzer is one of various Web analyzer PC-based products in the market-place. You can download this tool from your preferred download site or from the WebTrends Web site. You can try this product for 14 days. At the end of the trial period, you can buy this product or uninstall it.

WebTrends Analyzer analyzes only PC files of your log files in the AS/400 server. If your AS/400 server log files are located in a folder, you can directly map the location of log files to your AS/400 server directory using Client Access for Windows. Or, you can download the log files to your PC directory. If your log files are located in QSYS, which is in EBCDIC format, you should convert them to the ASCII format using the Copy to PC Document (`CPYTOPCD`) command. It copies your desired log files to the folder.

To analyze your AS/400 Web server, create your profile first. Go to **File->New Profile**, or press the Ins key. Since your Web server is a single server, click the **Next** button. On the title, URL windows, type a simple description of your profile and enter your log file. You can browse Network Neighborhood to find your AS/400 server. Make sure you already logged on through Operations Navigator if you want to directly map to your access log file inside the AS/400 server. If the log file is already located in your PC directory, type in the directory and file name. The AS/400 system creates one log file per day for each log type. To analyze log files in a certain time frame, you should combine all logs file that were created.

On Log File Format options, select **IBM Internet Connection Secure Server**. Do not allow the Auto-detect log file type because it will not work for AS/400 server log files.

Click the **Next** button. The Home page window appears. On the Home Page File Names Options, enter your default first page. You can refer to your server instance to get the file name of the first page. By default, the AS/400 system gives us Welcome.htm.

Specify the URL address of your AS/400 server. Do not put the Welcome or Index page on the Web Site URL options.

Click the **Next** button two times until you reach the Finish button. The rest of the options are not necessary, so leave them as they are. Click the **Finish** button.

You can add several profiles for several logs or a Web server. After you do that, the window shown in Figure 37 appears.

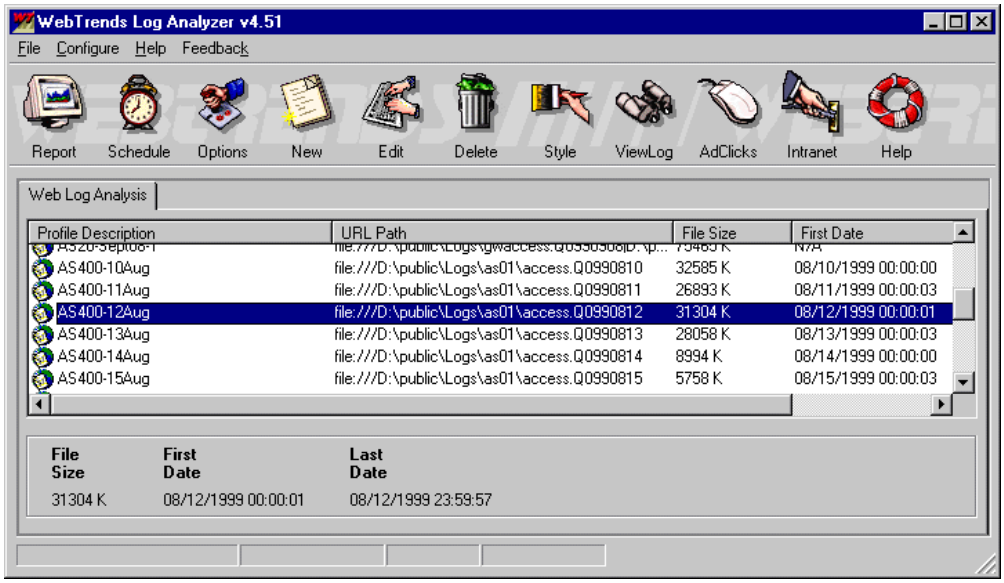


Figure 37. WebTrends Log Analyzer main menu with many profiles

Click the **Report** icon in the tool bar. Leave all of the options as they are. In the following sections, we discuss the relationship of the report to AS/400 server performance, and how to get it and relate it to Performance Tools.

Determining the most active time or hour

From the report Web pages, select **Activity Level by Hour** under the **Activity Statistic** section (Table 1).

Table 1. Activity Level by Hours Details to determine the hits and bytes transferred

Activity Level by Hours Details			
Hour	**# of Hits**	**% of Total Hits**	**# of User Sessions**
00:00-00:59	42,030	2.08%	2,624
01:00-01:59	52,009	2.58%	3,272
02:00-02:59	65,965	3.27%	3,528
03:00-03:59	72,573	3.6%	3,790
04:00-04:59	71,630	3.56%	3,791
05:00-05:59	62,736	3.11%	3,608
06:00-06:59	77,802	3.86%	4,358
07:00-07:59	109,986	5.46%	6,028
08:00-08:59	140,904	7%	7,330
09:00-09:59	131,874	6.55%	6,728
10:00-10:59	120,887	6.01%	6,447
11:00-11:59	119,844	5.95%	6,855
12:00-12:59	137,183	6.82%	6,473
13:00-13:59	127,720	6.34%	6,790
14:00-14:59	132,683	6.59%	6,774
15:00-15:59	120,049	5.96%	6,403
16:00-16:59	90,582	4.5%	4,838
17:00-17:59	60,884	3.02%	3,466
18:00-18:59	52,517	2.61%	2,860
19:00-19:59	47,857	2.37%	2,628
20:00-20:59	48,660	2.41%	2,731
21:00-21:59	47,823	2.37%	2,564
22:00-22:59	40,247	2%	2,374
23:00-23:59	36,928	1.83%	2,253
Total Users during Work Hours (8:00am-5:00pm)	**1,121,726**	**55.76%**	**58,638**
Total Users during After Hours (5:01pm-7:59am)	**889,647**	**44.23%**	**49,875**

In this report, such as numbers hits per day and hits per hour are important. Unfortunately, we cannot relate the number of users to the Performance Tools, because Performance Tools does not recognize the Web user. In fact, many "window shopping users" have accessed the Web server.

Using the data in another way, Web Analyzer also produces reports using bar graphs.

Determining the bytes transferred

You can open the **Summary of Activity by Time Increment** report under the **Activity Statistic** section to determine the bytes transferred by day. To get this report, generate a report for a one day log file. A report for one week or one

month displays different details. For a one-week report, the detail goes to the daytime interval, and so on.

Table 2. Activity report by day to determine hits and bytes transferred per day

Summary of Activity by Time Increment				
Time Interval	**Hits**	**Page Views**	**KBytes Transferred**	**User Sessions**
00:00:01 - 01:00:00	4,832	1,532	33,390 K	410
01:00:01 - 02:00:00	7,128	2,362	60,649 K	501
02:00:01 - 03:00:00	9,629	3,358	118,747 K	567
03:00:01 - 04:00:00	10,929	4,353	118,945 K	577
04:00:01 - 05:00:00	11,536	5,023	151,731 K	556
05:00:01 - 06:00:00	11,300	4,827	118,692 K	563
06:00:01 - 07:00:00	13,157	5,514	135,297 K	711
07:00:01 - 08:00:00	18,833	7,293	172,118 K	956
08:00:01 - 09:00:00	24,211	8,906	307,779 K	1,222
09:00:01 - 10:00:00	24,232	8,261	258,849 K	1,276
10:00:01 - 11:00:00	24,068	8,039	291,957 K	1,244
11:00:01 - 12:00:00	18,842	6,454	224,678 K	1,113
12:00:01 - 13:00:00	21,714	8,768	211,517 K	951
13:00:01 - 14:00:00	20,334	6,816	194,229 K	1,091
14:00:01 - 15:00:00	22,150	6,986	212,067 K	1,062
15:00:01 - 16:00:00	19,401	6,457	187,972 K	987
16:00:01 - 17:00:00	15,517	4,885	140,163 K	808
17:00:01 - 18:00:00	9,852	3,271	141,416 K	514
18:00:01 - 19:00:00	8,634	2,806	116,309 K	422
19:00:01 - 20:00:00	6,895	2,107	50,930 K	440
20:00:01 - 21:00:00	7,160	2,445	59,206 K	397
21:00:01 - 22:00:00	6,920	2,721	85,209 K	381
22:00:01 - 23:00:00	4,697	1,759	45,142 K	336
23:00:01 - 00:00:00	4,420	1,583	34,091 K	349
Total	**326,391**	**116,526**	**3,471,083 K**	**17,434**

Table 2 helps to determine the busiest times. The bytes transferred and the hits are the most important data here. The page view is less important, but it gives you a sense of the pages that are accessed per day. Bytes and hits are important because they are related to Performance Tools.

It is easier to represent the data in graph reports, as shown in Figure 38.

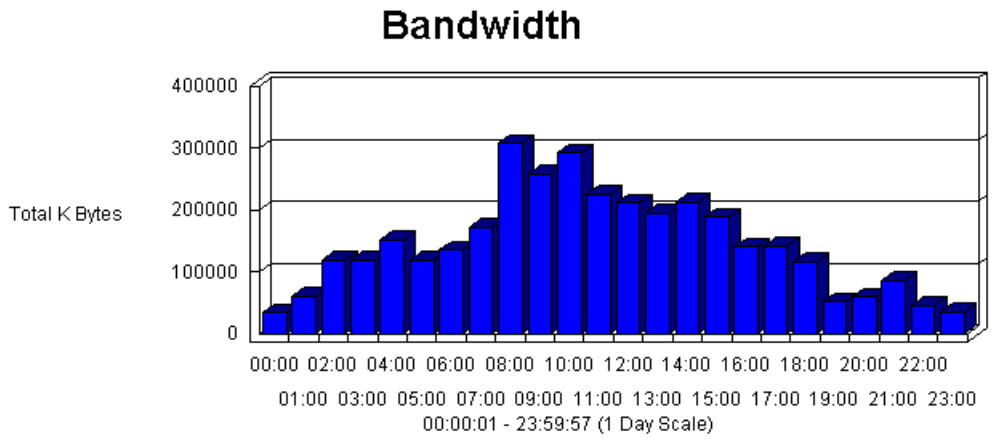


Figure 38. Bar graph report for bytes downloaded per hour

Based on the graph in Figure 38 on page 59, we can analyze whether the busiest time happens when the most resources were occupied, the CPU response time, IOP utilization, whether main memory page fault happened, disk arm utilization, and so on.

To correlate this bar graph, you can choose the Time Interval on your Performance Tools Report according to the most busiest time.

Determining the number of hits per day
Open the **Technical Statistics and Analysis** report (Table 3) in the **Technical Statistic** section.

Table 3. Total hits and cached hits report

Technical Statistics and Analysis	
Total Hits	2,048,957
Successful Hits	2,011,373
Failed Hits	37,584
Failed Hits as Percent	1.83%
Cached Hits	422,008
Cached Hits as Percent	20.59%

The number of hits and the number of successful hits compared to the number of failed hits gives us the percentage. Of course, we want 100% successful hits to give our users the best services. From this report, we can determine what happened to our Web server. Unsuccessful hits can occur due to a lack of network bandwidth, CPU utilization, page faults, and so on.

If you enable the cache capability on the AS/400 server, this report gives you the percentage and total cache hits. Cache hits improve AS/400 Web server performance. If there are too many cache hits and your main AS/400 main memory is too small, a memory problem can occur on your server.

Determining the most downloaded file type
Click on the **Resources Accessed** section. Then, click on the **Most Downloaded File Types** report (Table 4).

Table 4. Most downloaded file types

Most Downloaded File Types			
	File type	**Files**	**K Bytes Transferred**
1	gif	748,976	3,517,530
2	htm	644,594	8,512,362
3	css	60,298	86,536
4	html	52,114	827,881
5	jpg	45,014	741,231
6	js	14,584	145,185
7	d2w/input	7,124	91,277
8	rpm	3,803	273
9	pdf	3,419	2,789,923
10	d2w/report	1,947	23,420
	Total Files & K Bytes Transferred	**1,581,873**	**16,735,614**

This report is important for determining whether users are using FTP sessions (if you open an FTP server). If FTP access is dominant, Performance Tools may not

represent the performance perfectly. Because FTP traffic is not recorded by the Web log analyzer, the bytes are transferred in Performance Tools and will be higher than the Web log report.

Files, such as .gif, .htm. or html, .jpg, and .js, are mostly downloaded through HTTP protocol. However, files, such as .pdf, .exe, and .zip files, are perhaps downloaded through FTP.

Determining the dynamic pages and forms

Open the **Dynamic Pages & Forms** report (Table 5) under the **Resources Accessed** section.

Table 5. Percentage of dynamic pages accessed

Dynamic Pages & Forms				
	Dynamic Pages	No. of Pages	% of Total	User Sessions
1	http://www.myserver.com/cgi-bin/as400src/	23,132	40.69%	12,752
2	http://www.myserver.com/servlet/feedspd	22,101	38.87%	254
3	http://www.myserver.com/net.data/rmedia/rmosverf.d2w/input/	2,794	4.91%	802
4	http://www.myserver.com/net.data/rmedia/rmosinfo.d2w/input/	2,248	3.95%	785
5	http://www.myserver.com/net.data/rmedia/rmoshome.d2w/report/	1,918	3.37%	780
6	http://www.myserver.com/howtobuy/ShowChoices/	802	1.41%	732
7	http://www.myserver.com/e-solution/Processesolution/	618	1.08%	403
8	http://www.myserver.com/QSYS.LIB/SNIPPETS.LIB/sendfile.pgm	542	0.95%	234
9	http://www.myserver.com/net.data/rmedia/rmoshome.d2w/input/	419	0.73%	262
10	http://www.myserver.com/myas400/SIGNON/	383	0.67%	234

Dynamic pages have different characteristics than the static pages. Unfortunately, this report does not give the bytes that were transferred and the hits that occurred. To understand the performance, you can see the most programs that were accessed. You can trace them by using the Performance Trace Report. This report shows the resource utilization and time consumed on each transaction. Then, the total amount of pages are calculated based on the assumption that one program generated one page result.

3.5.3.2 NetIntellect

This is another tool with features that are available to present a report from the AS/400 Web server. NetIntellect is a product of Webmanage Technologies, Inc. NetIntellect offers you the ability to choose different reports from the main screen (Figure 39 on page 62).

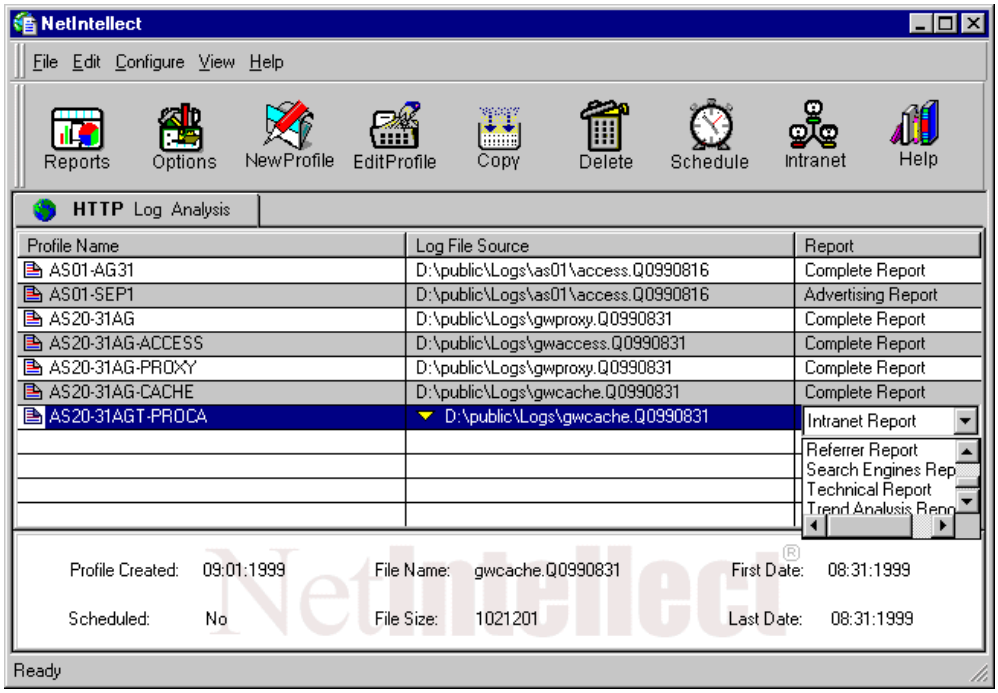


Figure 39. NetIntellect screen to select different report type

This software needs an additional add-on file to include several log files in one report. If you need to generate a report of many days worth of log files (because the AS/400 server creates one log file per day), you need the add-on software. Or, you can append a copy from many log files to one new log file.

Basically, this tool performs the same reports as WebTrend Analyzer. For example, the graph in Figure 40 reports daily activities.

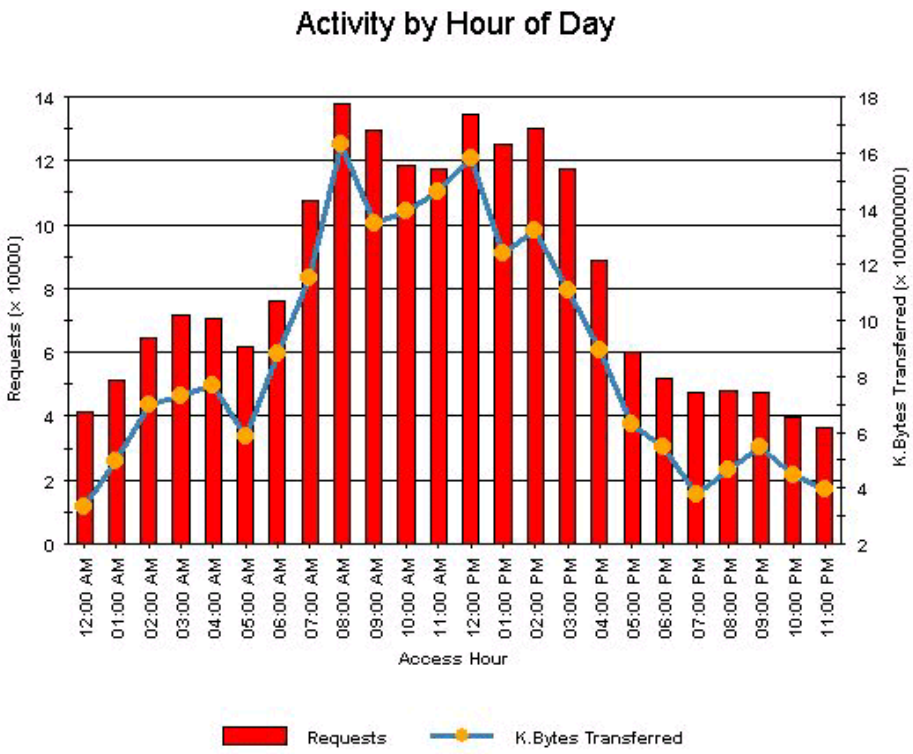


Figure 40. Activity bytes and request per day

The analyzer you choose is up to you.

Using the same method described on 3.5.3.1, “WebTrends Analyzer” on page 56, you can determine the information that you need and correlate it to Performance Tools for AS/400.

Chapter 4. Correlation of measured performance data

Performance measurement is about monitoring in accordance with a service level agreement. It can be further subdivided into two areas for the AS/400 server: performance monitoring and performance reporting.

There are four aspects that need to be measured from a performance point of view for any computer system. These are:

- CPU resource
- Memory
- Disk arms
- Communication utilization

This chapter discusses the relationship between Web performance measurements and AS/400 system resources measurements.

4.1 Measuring AS/400 system resource usage

AS/400 Web servers log all Web activities regarding the setting from Web server instances. This log runs all day and is generated early the next day, starting at midnight. Although this is not the AS/400 server resource that we want to measure, we cannot neglect this report.

While the Web server is active, you must run Performance Tools to track the AS/400 resources. Using Performance Tools for AS/400 can assist while analyzing, reporting, and modeling in managing the performance of the AS/400 system as a Web server.

In this redbook, we do not cover how to run and setup the Performance Tools for AS/400. However, we show you the important parameters that you should choose for optimal data.

4.1.1 Analysis measurements

For the purpose of analysis, the time interval should be as small as possible. The smallest number is 5 minutes for the INTERVAL option, as shown in Figure 41 on page 66. It should run the entire day. Set the option ENDTYPE = *ELAPSED. Specify 24 hours. The next 24 hours of performance collection data will end automatically. It is ideal if it is started early in the day according to the Web log period time of recording. Or, you can choose to stop the Performance Monitor at a certain time. Set Stop collection options = *TIME, and then set hours = 0, minute = 0, and second = 0 to stop at midnight.

Actually, activating TRACETYPE = *ALL impacts the entire AS/400 system performance. Because the trace generates data every 0.5 seconds, the system is busy servicing the trace. This option is useful for creating the Transaction Report later.

```
                Start Performance Monitor (STRPFRMON)

 Type choices, press Enter.

 Member . . . . . .   WEBPFR19AG    Name, *GEN
 Library  . . . . . > HTTPPERF      Name
 Text 'description' . . . . . .   Web Performance Data Mntr for Aug 19,
1999.

 Time interval (in minutes) . . . > 5             5, 10, 15, 20, 25, 30,
35...
 Stops data collection  . . . . .  *ELAPSED     *ELAPSED, *TIME, *NOMAX
 Days from current day  . . . . .   0           0-9
 Hour . . . . . .   24           0-999
 Minutes  . . . . . .   0           0-99
 Data type  . . . . . .   *ALL         *ALL, *SYS
 Select jobs  . . . . . .   *ALL         *ALL, *ACTIVE
 Trace type . . . . . .   *ALL         *NONE, *ALL
 Dump the trace . . . . . .   *YES         *YES, *NO
 Job trace interval . . . . . .   .5          .5 - 9.9 seconds
 Job types  . . . . . .   *DFT        *NONE, *DFT, *ASJ, *BCH...
              + for more values
                                                                More...
```

Figure 41. Starting Performance Monitor for Web performance analysis

4.1.2 Sizing measurements

For sizing and modeling purposes based on current resources and the current workload, and then extrapolating to the expected or predicted workload, the main difference between sizing measurements is highlighted on the peak activity of the entire report.

To determine which is the peak activity of a certain time period, with respect to the Web analyzer, we should get the most active hour of user hits and the number of bytes transferred in that total period.

4.2 Correlation of Web hits to AS/400 system resources

Web hits are activities that impact AS/400 system resources. The number of hits represents the number of activities, and the AS/400 system as a Web server gives the service as requested. To perform the requested service, the AS/400 system uses all available resources.

This section discusses general AS/400 resources that directly impact the user's hits.

4.2.1 Correlation of jobs to hits

If you create a server instance, your HTTP server job automatically creates with the same name of your server instance. HTTP server jobs are considered batch jobs. They are a predefined group of processing actions submitted to the system to be performed with little or no interaction between the user and the system. A batch job is typically a low priority job and can require a special system environment in which to run.

TCP/IP jobs, in general, like other jobs on the AS/400 system, are created from job descriptions and associated classes. The job descriptions and classes should be adequate in most cases.

One server instance may create many jobs with one thread in different job numbers. The reason for having multiple server jobs 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, for N-way systems, each CPU may simultaneously process server jobs. The system adjusts the number of servers that are needed automatically (within the bounds of the minimum and maximum parameters).

These jobs bring you to determine the amount CPU utilization, the number of disk I/O transactions that will happen, and so on.

If you look at the Component report under Job Workload Activity heading, the column is not entirely useful for HTTP performance. HTTP server traffic is never calculated as a transaction of a job. The Transaction or Transaction per Hour column only applies to interactive jobs, not to the HTTP server, which is batch job (Figure 42).

Job Name	User Name/ Thread	Job Number	T y p	Pl	P t y	CPU Util	Tns	Tns /Hour	Rsp	Disk I/O Sync	Disk I/O Async	Disk I/O Logical	Cmn I/O	PAG Fault	Arith Ovrflw	Perm Write
GEORGE	QTMHHTTP	057731	B	02	25	14.082	0	0	.000	68	0	6606	0	0	0	
GEORGE	QTMHHTTP	057789	B	02	25	.000	0	0	.000	0	0	0	0	0	0	
GEORGE	QTMHHTTP	057833	B	02	25	.028	0	0	.000	0	0	0	0	0	0	
GEORGE	QTMHHTTP	057865	B	02	25	.003	0	0	.000	0	0	0	0	0	0	
GEORGE	QTMHHTTP	057880	B	02	25	.000	0	0	.000	1	0	0	0	0	0	
GEORGE	QTMHHTTP	058205	B	02	25	.000	0	0	.000	0	0	0	0	0	0	

Figure 42. Job Workload Activity report

Synchronous disk I/O operations occur when the job that issued a disk I/O request waits for the operation to complete before continuing. For example, a database request that was submitted from Web server read by key is a synchronous operation, and job processing waits until the data is in main storage.

Asynchronous disk I/O operations refer to operations where the job does not necessarily wait for the disk I/O operation to complete. It allows the job to continue processing while the operation is in progress.

Logical disk I/O operations occur when a buffer is moved between the user program and system buffers. Often, it can be satisfied by data already residing in memory.

If numbers request a database from a Web server to an AS/400 system, for example, using Net.Data or a CGI-bin program, synchronous disk I/O activities will increase.

If the AS/400 main storage pool is large enough to load most all of the Web pages, logical disk I/O activities will be more dominant than asynchronous disk I/O activities.

4.2.2 Correlation of CPU utilization to hits

The number of hits represents the user activities. The correlation between the number of users and the number of hits per day for marketing purposes is a helpful indication in analyzing the trend of Web user activity. An example report is shown in Table 6. For example, in one month, the Web site changed four times. The change was made every week. If the number of hits per user is tremendously high, it can be concluded that the users like the content.

Table 6. Hits per day and KBytes transferred report

Summary of Activity by Time Increment				
Time Interval	Hits	Page Views	KBytes Transferred	User Sessions
08/10	338,769	124,927	3,694,576 K	17,907
08/11	283,601	91,661	2,956,030 K	16,911
08/12	326,391	116,526	3,471,072 K	17,369
08/13	289,334	107,600	2,781,932 K	14,959
08/14	87,992	26,833	704,029 K	4,621
08/15	60,274	20,614	546,709 K	4,484
08/16	324,187	116,974	3,307,981 K	17,218
08/17	301,025	127,073	3,0701,766 K	15,044
Total	**2,011,373**	**732,208**	**20,533,095 K**	**108,513**

We want to correlate hits per user to AS/400 system resources. Because some native log files are only generated early in the day for the log of the previous day, on a single day we can get one single log by default. Performance Tools can run in a certain window of time, according to the log file generated. To correlate it, we run Performance Collection Data for the entire day. Then, we can observe the statistics of the correlation between hits per user per AS/400 server resources.

The CPU Utilization number comes from the System Report of Performance Tools database. An AS/400 Web server mostly uses priority number 25, and the type of job is a batch job. Under Non-Interactive Resource Utilization Expansion, you can see the detail of each Priority report and Job Type report. CPU Utilization Average is the cumulative number of all job utilizations, such as System Job, DDM Server Job, Client Access Job, PassThru Job, and so on.

Instead of comparing utilization for a period of days (one week, one month, and so on), you can divide the utilization into certain time intervals, for example, per one hour of access time. You can print a Performance Report for every time interval option you need.

You can see a correlation of the data in Figure 43 on page 69.

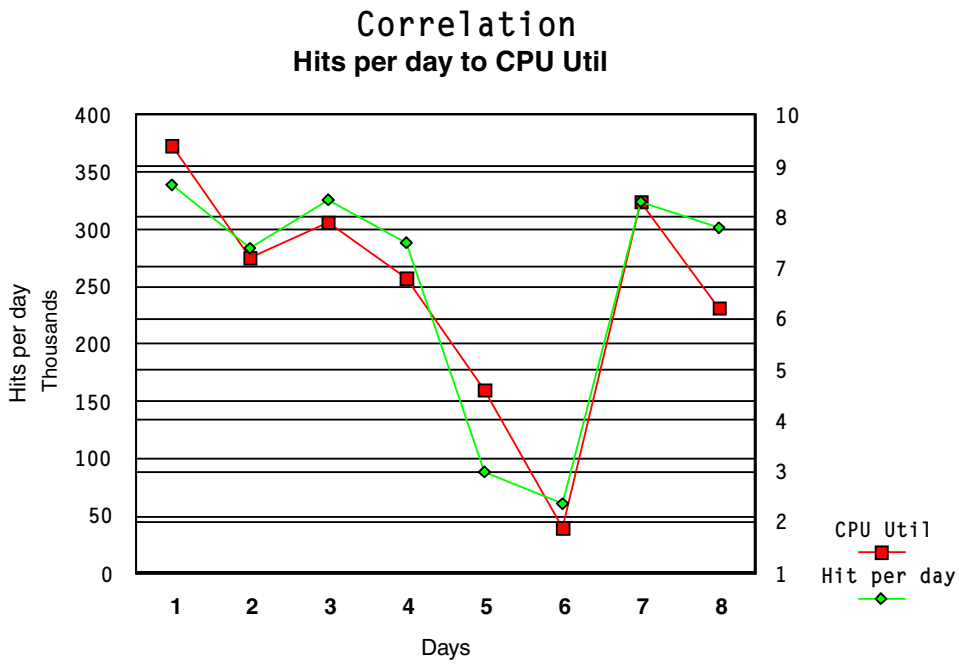


Figure 43. Correlation hits per day to CPU utilization

Some days, CPU utilization does not show a linear correlation to the hits per day number. If CPU utilization tends to be higher compared to the average CPU utilization in the amount of hits, there may be many transactions that utilize CPU resources, such as database transactions, CGI-bin, Net.Data, and RPG program.

You can see the trends that the Dynamic Pages & Forms report maps to the differentiation of CPU Utilization.

Table 7 shows an example of a one day report. You should create the series of reports and put it in a spreadsheet of CPU utilization. Concentrate on the trend of the cached percentage to CPU utilization.

Table 7. Total hits per day and cached hits report

Technical Statistics and Analysis	
Total Hits	89,380
Successful Hits	87,792
Failed Hits	1,588
Failed Hits as Percent	1.77%
Cached Hits	12,666
Cached Hits as Percent	14.17%

If CPU utilization tends to be lower than average, which should happen for certain hits per second, you should be aware of the percent of cached hits. A higher percentage of cached pages and a lower number of hits per day generate lower CPU utilization. The lower cached pages percentage and the higher number of page hits per day impacts higher CPU utilization.

4.2.2.1 CPW correlation to hits per second

Hits per second is a commonly used metric of Web server performance (see 1.3, “Commonly used metrics for Web applications” on page 12). Theoretically, higher CPW capacity can handle more hits per second. Depending on the content of the

Web page, the capacity varies. For sizing purposes, IBM has standard hits per second per CPW numbers for certain Web page types.

4.2.3 Main memory and system pool

The effect of memory demand can be observed and measured to a certain degree by using page faulting rates in memory pools. Page faulting happens if a particular job requests data that is not already in the storage pool. It must retrieve it from disk. A page fault is a notification that occurs when an address that is not in main storage (memory) is referenced by an active program.

To determine which pool is used by your Web server, you should first remember the server instance that you are using. For example, the server instance is WEBSVR. On WEBSVR line, you will find User Name/Thread, Job Number, Type, Pool, Priority, CPU Util, and so on. Write down the pool ID that the job ran in, for example, Pool ID number 2.

Open the **System Report** under the **Storage Pool Utilization** section. Pay attention to Pool ID number 2. This is the pool ID in which the Web server is run. See Figure 44 for an example.

							---------- Avg Per Second ---------				---- Avg Per Minute -----		
Pool ID	Expert Cache	Size (K)	Act Lvl	CPU Util	Number Tns	Average Response	------ DB ------ Fault	Pages	---- Non-DB ---- Fault	Pages	Act- Wait	Wait- Inel	Act- Inel
01	0	272,520	0	4.5	0	.00	.0	.0	.4	5.4	5	0	0
02	3	2,293,884	184	14.7	156	.01	.0	.0	.7	3.9	1,389	0	0
03	0	28,832	4	.0	0	.00	.0	.0	.0	.0	1	0	0
04	0	288,348	125	.0	0	.00	.0	.0	.0	.0	0	0	0
Total		2,883,584		19.3	156		.0	.0	1.1	9.4	1,397	0	0

Figure 44. Storage pool utilized by Web server

The use of main storage by a job cannot be directly measured with any of the available performance tools. Nor is it possible to determine how much memory a particular job is using.

We cannot directly relate the hits to the storage pool measurement. The direct relation to this measurement is CPU utilization. Understand that CPU utilization is directly related to the number of hits, so storage pool utilization is indirectly correlated to the number of hits.

Let us see the Average Per Second column either under database (DB) or Non-database (NDB) Pages or Fault. In this example, NDB fault is only 0.7 per second, or in 1.4 second it will be 1 NDB fault.

A database fault occurs when referencing database data or access paths. A non-database fault refers to objects other than database objects. NDB pages include programs, data queues, configuration objects, internal space objects, and so on. If the NDB page fault rate in the machine pools (pool 1 or *MACHINE) is greater than 5 to 10 faults/sec on a system, performance may be improved by increasing the size of the machine pool, therefore reducing the page fault rate. The only way to control the fault rate in the machine pools is to change the size of the pool. You cannot increase or decrease the activity level of the *MACHINE pool.

When either type of fault occurs, the job must wait for the necessary information to be transferred from disk to main storage. A disk operation that the job must wait for is called *synchronous disk I/O*.

The average response time per second can be related to the process of HTTP traffic requested from the client, accepting the request by the server, until it is shown to the browser. For more information about the requests between the client and the server, and for details about Figure 16 on page 20, refer to 2.1, “Basic Web serving” on page 19. On the server side tasks, the average response time can be assumed as the server reads from memory or disk tasks.

In one page, a request may not only access memory or disks one or two times, but many times. If the Web server accesses main memory, this number is qualified. In some cases, you do not know where the Web server gets the data, because AS/400 systems implement single level storage. It may be from existing memory, library files, IFS, source files, database files, the QDLS share folder, or other system files available in the AS/400 system.

If you set the AS/400 Web server to cache the Web pages, cache is allocated to the main memory. Cache and proxy performance increase according to the additional memory that is available.

4.2.4 Disk arms

You can obtain the Disk Utilization Report from the System Report in Performance Tools. The percentage that a disk arm is busy is the key disk measurement for evaluating disk performance. You can see this in the Percent Util (average disk operation utilization) column of the report. You can also see the disk I/Os per second under the Op per Second column heading. The Op per Second report contains the values of the physical disk operations issued by the SLIC.

If the utilization becomes higher than the average of 40% per disk arm for all disks (in multiple disk arm system), the queuing on each disk arm begins to degrade performance. For a comparison, compare the individual percentage utilization to the average percentage utilization. If one or more disks are busier than average, determine the usage by using the Performance Monitor to collect disk I/O operations data.

In most cases, when the AS/400 system acts as a Web server, disk activity is utilized by other commercial applications that run in the same AS/400 system. In commercial applications, disk activity is a major part of the overall response time. If Web applications are integrated into commercial applications, the degradation of performance may occur due to commercial applications accessing the disk, instead of Web server access to disks.

Paging activity contributes to (but is not necessarily all of) the NDB read count. A high NDB read count in a job (assuming HTTP server jobs) may be caused by the way the job accesses data, the program structure, or the use of program working storage.

If the storage pool that is selected to run Web server application jobs is not large enough, excessive paging can occur. This directly affects performance on the AS/400 system and the performance of the applications.

The correlation of disk activity to the Web server is mostly on the performance of paging from disk to main memory.

4.2.5 Transaction trace

For additional analysis, you may need to set the collection performance data to the option TRACE(*ALL). This option collects data that enables you to see which job, program, or transaction is causing the heaviest use of resources, or to simply measure the resource usage by the job, program, or transactions while they are being run or submitted.

Trace data is used for the Transaction Report and Lock Report output. Transaction report data includes much more detail per job regarding sign on and sign off, response time, CPU seconds, disk I/O operations, object lock/seize times, time slice end, short wait times, and so on.

Note

Performance Tools trace jobs require excessive amounts of CPU during an interactive transaction. Do not produce the detailed job analysis until you identify the programs or jobs that you want to analyze.

The summary reports allow you to determine the overall performance of the job without analyzing the Detail report.

To print the Performance Report with Trace options, submit the command:

```
GO PERFORM
```

Enter option `3` (Print Performance Report). Then, enter option `3` (Transaction Report). On the option Report Type (RPTTYPE), enter `*TNSACT` and `*TRSIT`.

To minimize the report size, you can only select the job that is running as an HTTP server. Normally, the job name is synchronous to the HTTP server instance name.

You get two kinds of reports: the Transaction Report and the Transition Report. From these reports, you may get the following information to analyze job performance:

- Programs called and the calling sequence and frequency
- Wall clock time of the program call and return sequence
- CPU time used by each program
- The number of synchronous DB and NDB disk I/Os per program called
- The number of full and shared file opens
- Messages received by each program

To analyze the reports, you have to open Component Report print, Transaction Report print, and Transition Report print.

From the Component Report, you can see the job number that occupies the most resources, such as CPU utilization and disk I/O transaction. It depends on your reasons for tracing. For example, it may be important for you to know CPU utilization, or maybe Disk I/O transaction information is more important. In this report, you cannot specify what the AS/400 system has done, the number of

resources that were occupied, the time the transaction started and ended, the number of synchronous or asynchronous I/Os that were accessed, the program names were active, and so on. However, you will know which job number is the most active.

The Transaction Report output has two parts:

- The details, which show data about every transaction in the job
- The summary, which shows data about the overall job operation, normally located in the bottom of each report section

If there are response times that are not acceptable compared to objectives, you should read the report further.

If you want to know all of the state changes within a transaction, you should review the Transition Report. This report is composed of the following two sections:

- Transition detail, which shows each state transition made by the job
- Summary, which shows the same data as the summary output from the Transaction Report

Figure 45 shows an example of the Transition Report.

```
Job name  . . : GEORGE     User name . . . :       QTMHHTTP     Job number . . . :   059441   TDE/Pl/Pty/Prg . :    F757/02/25/YE
Thread  . . . : 0000004D
Partition ID  : 00         Feature Code  . :2388
Job type  . . : B       Elapsed Time -- Seconds            Sync/Async Phy I/O    -MPL-
                        ----------------------          ---------------------- C   I     Last 4 Programs in Invocation Stack
             State Wait   Long   Active   Inel    CPU    DB    DB  NDB   NDB      u   n ------------------------------------------
    Time     W A I Code   Wait    /Rsp*   Wait    Sec    Read  Wrt Read  Wrt  Tot r   l    Last      Second      Third     Fourth
------------ ----- ---- ------- ------- ------- -------- ---- ---- ---- ---- ---- -- -- ---------- ---------- ---------- ---------
13.37.32.862 ->A           .008                                                  10
13.37.33.974   W                  1.111            .014    1        15            9    QC2IFS     QZHBSUTL   QZHBSUTL   QZHBSUTL
13.37.34.821   W                   .848            .010    1         8            5
13.37.35.189   W                   .367            .006    1         4            8
13.37.38.123   W                  2.935            .036             27            8
13.37.39.123   W                  1.000            .011              8            7
13.37.41.307 W<-                  2.184            .003              3            7    QSOSRV1    QZHBSUTL   QZHBSUTL   QZHBSUTL
 ----------  QZHBSUTL             8.445*           .080    3    0   65    0   68*
```

Figure 45. Transition Report

The job name under GEORGE is based on the HTTP server instance. The Job number is 059441. You can go directly to this job number after looking at the Component Report, which was the most active or takes the most resources, depending on your purpose.

The total CPU time is 0.080 seconds. During this time (from 13.37.32.862 to 13.37.41.307), the DB Read is 3 and NDB Read is 65. For example, if there are 200 DB Reads (database read operations) per transaction, the response time is unacceptable.

4.3 Correlation of Web hits to network resources

The number of hits to the Web server cannot be measured using Performance Tools. As stated in Chapter 3, “Performance measurement and analysis tools” on page 27, the Performance Tools database does not log TCP/IP traffic. Correlation between the number of hits and network resources can be measured using log access of the Web server instance. You can analyze it using log analyzer

software. The reason for using log analyzer software is to calculate the total number of bytes transferred.

NetIntellect only shows you the total bytes, but WebTrends Analyzer shows you the number of hits, the number of accessed pages, and the number of users. For example, if you are using WebTrends Analyzer, select **Summary of Activity by Time Increment** under the **Activity Statistic Section**. The data shown in Table 8 will appear.

Table 8. Summary of daily activity by time increment

Summary of Activity by Time Increment				
Time Interval	**Hits**	**Page Views**	**KBytes Transferred**	**User Sessions**
08:00:00 - 08:59:59	175	22	855 K	1
09:00:00 - 09:59:59	861	431	3,755 K	0
10:00:00 - 10:59:59	3,840	528	23,446 K	0
11:00:00 - 11:59:59	5,629	644	34,163 K	2
12:00:00 - 12:59:59	6,268	683	36,730 K	0
13:00:00 - 13:59:59	0	0	0 K	0
14:00:00 - 14:59:59	1	0	30 K	1
Total	**16,774**	**2,308**	**98,979 K**	**4**

In a different time interval, you can create different Performance Monitor data for each individual time interval, which may appear as shown in Figure 46.

```
                      Print Performance Report

Library . . . . . .   QPFRDATA

Type option, press Enter.
  1=System report   2=Component report   3=Transaction report   4=Lock report
  5=Job report   6=Pool report   7=Resource report   8=Batch job trace report

Option   Member       Text                                 Date       Time
  2      Q992431159   HTTP_PERF                         08/31/99   11:59:06
         Q992431119   HTTP_PERF                         08/31/99   11:19:29
         Q992431016   HTTP_PERF                         08/31/99   10:16:30
```

Figure 46. Print Performance Report of Performance Tools data

Make individual printouts. For example, make a printout for member `Q992431159` whose performance collection ended on 11:59:06. It correlates with line number 5 of the Summary of Daily Activity by Time Increment report.

Print the Component Report, and select IOP Utilization. A report appears that looks like the example in Figure 47.

```
                    --- IOP Processor Util ---    DASD     -- KBytes Transmitted --   Available
IOP                 Total   Comm   LWSC   DASD   Ops/Sec       IOP        System       Storage
------------------- -----  -----  -----  -----  ----------- ----------- -----------   -----------
CMB01        (6757)   6.1     .0     .1    2.4                   7,520       52,689    25,783,691
CMB02        (2809)   1.1     .0     .0     .0                       7            0    15,591,232
CMB03        (2809)   4.1    4.0     .0     .0                  32,048        9,148    13,599,744
```

Figure 47. IOP Utilization report

We use the Integrated Netfinity Server (INS) at MFIOP CMB03 and do not activate the INS as a server. We only use the Token-Ring card to connect to the isolated LAN and dedicated TCP/IP traffic.

4.3.1 Bytes per second per IOP traffic

Here is an example of bytes per second that were transferred based on the System Report from the Performance Report:

```
Received        Transmitted
----------       ---------
1092.6          9256.1
```

The average bytes received per bytes transmitted is 10%, in every Web transaction. In this example, 90% bytes is transmitted to the client, and 10% of the bytes are transmitted to the server. For more details about the kind of traffic that is involved here, refer to Chapter 2, “Characteristics of Web-based applications” on page 19.

4.3.2 Correlation to the bytes transmitted

In Figure 47, refer to the IOP CMB03 row and the KBytes Transmitted column. The total bytes transmitted is close to the total KB transferred in the log analyzer. In the IOP point of view, total KB transmitted consists of sending and receiving bytes. That is why the total traffic bytes passing through the IOP is more than the KB transferred as recorded by the log analyzer. The bytes that are received and sent are not the same amount. This is called the *asynchronous transfer rate*. In this case, the comparison between the bytes captured as Web traffic to the total IOP bytes traffic is 89%.

The IOP Processor Utilization shown in Figure 47 is quite low. The recommended number is below 35% utilization and categorized as good. For another measurement, we find correlations (Figure 48 on page 76).

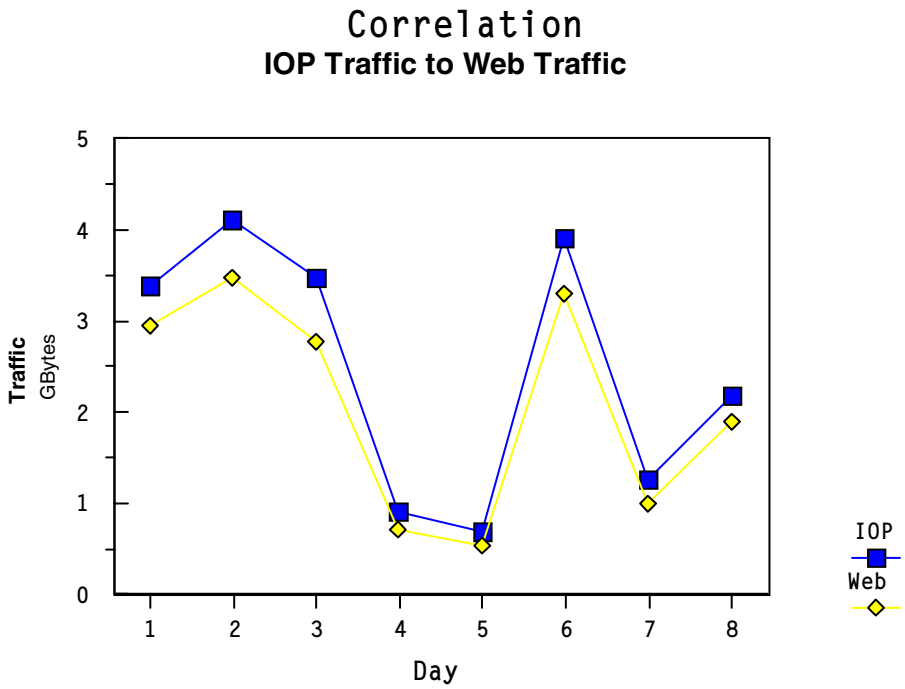


Figure 48. Graphic correlation of IOP traffic to Web traffic

In correlation between IOP traffic and Web traffic, Web traffic counted is always IOP traffic. With this card, it was proved that since MFIOP only allowed TCP/IP traffic to come through, the IOP utilization can be monitored from the Performance Report. If a large amount of bytes are transferred (for example, Day 1, 2, 3, and 6), there are many differences between IOP bytes transferred to the Web bytes transferred. To learn the details of the traffic in there, we should look at the details of jobs. We should also suspect that other TCP/IP traffic (such as FTP, which cannot be captured by Web analyzer) has been created by the users.

To find the differences in traffic, consider the following reports or charts:

- The *Most Downloaded Files*. Files, such as .htm or .html, .gif, .jpg, and .js files, can be downloaded through the HTTP protocol. Consider an example where you put downloadable binary files (.exe, .zip, .pdf) on your pages, open the FTP daemon, and allow users to access the FTP server as anonymous users. The percentage of binary downloadable files is significant. In this case, you can predict that most traffic comes from FTP access.
- The *Most Accessed Directories*. Sometimes you need to put binary downloadable files in certain directories. With the same analysis mentioned in the previous point, you can suspect what happens in the Web server.
- If the AS/400 server is enabled as a cache server or proxy server with the intention to accelerate Web access, FTP files are cached after the first time that they are accessed. The next request user is serviced from cached files for the same file services. In this case, the second access is serviced through the HTTP protocol, instead of the FTP protocol.
- The percentage of Error Pages Accessed to all of the traffic generates the traffic, but does not capture it as Web traffic. It is not too significant since your site design is good.

If the differences were around 10%, it can be assumed that there were bytes sent to the server.

For more details about the traffic through IOP, you can use a communications trace. However, this data is not related to IOP performance.

4.3.3 Correlation to IOP utilization

If you want a correlation between IOP utilization and either Web traffic or IOP traffic, you can find the linear correlations (Figure 49).

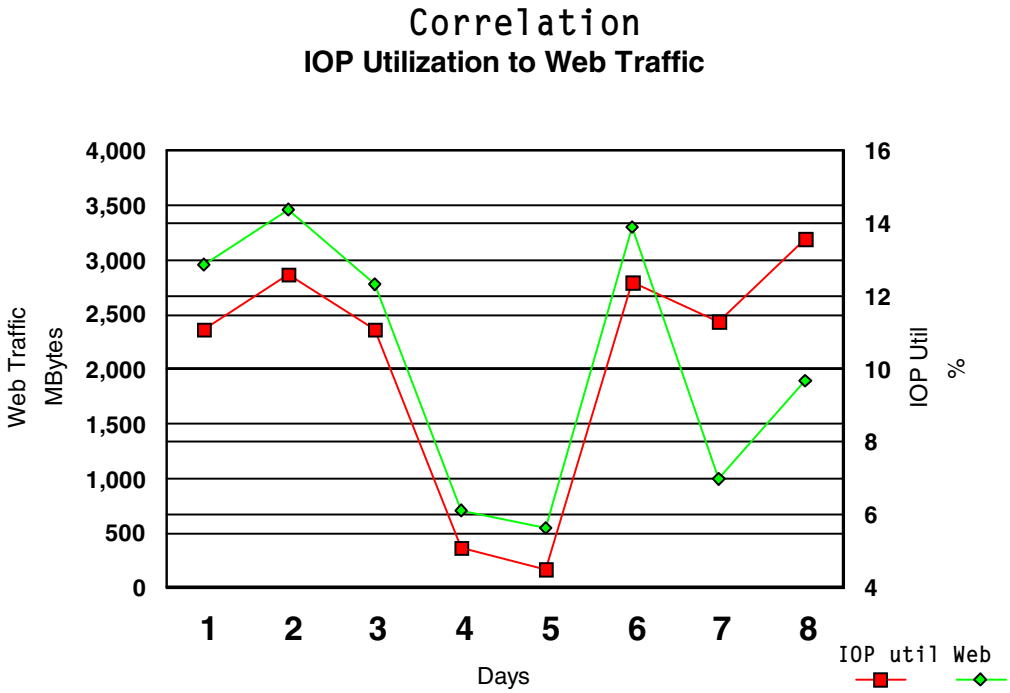


Figure 49. IOP Utilization to Web bytes transferred

If you put TCP traffic only in a Token-Ring card or put option *TCPONLY in an Ethernet card, it is easier to see the correlation between IOP utilization and bytes transferred due to the Web traffic.

The correlation between IOP utilization and Web bytes transferred is linear. If the standard IOP utilization allowed is 35% (see Appendix A in *AS/400 Performance Management V3R6/V3R7*, SG24-4735), 35% of the IOP utilization transfers about 9 GB per day. This number comes from the estimation of linear correlation between IOP utilization and bytes transferred:

```
9GB / (24 * 60 * 60) = 104.2 KB per sec.
or equal to 104.2 * 1024 = 106.7 Mbps.
```

In this example, 106.7 Mbps means that you need to add more network cards, because the capacity of the Token-Ring is 16 Mbps and the Fast Ethernet is 100 Mbps. Both do not consider the effectiveness of the network card. In other words, if you need to keep IOP performance healthy when the most traffic occurs, you need to add another network card.

4.4 Correlation of Web hits to client activity and client resources

In this section, *client* refers to where the browser was installed to access the Web server. Client resources are related to the CPU usage, memory usage, and whether there is available network usage.

Unfortunately, it is only possible to measure the PC resource from the Windows NT operating system (for Windows-based operating systems). In Windows 95, you should install another product to detect it. Another IBM PC operating system, such as Linux with KDE desktop installed, or GNOME, or another desktop management system can also measure the client resources using Task Management or Process Management. A UNIX workstation, such as the RS/6000 with the AIX operating system, has the same capability.

In a Windows NT workstation with Pentium II 350 MHZ processor and 128 MB memory, Netscape Navigator 4.6, by default, needs to use an average of 8.8 MB of memory and only one second of CPU time while idling. Microsoft Internet Explorer 5, by default, needs an average of 9.5 MB of memory usage and two seconds of CPU time while idling. Both conditions may differ for your PC configurations.

To open large static Web pages, the CPU time increases to about 10 seconds, and memory usage increases to 9.8 MB.

Simple dynamic pages increase CPU utilization to 8 seconds. Accessing large dynamic Web pages, such as searching through Net.Data, consumes higher CPU time to 16 seconds and increases memory usage to about 10 MB.

Opening pages with many graphics increases the CPU percentage to about 30% to 40%.

Resources are most demanding if the user insists on keeping the browser open to the previous pages by using the Back button in the browser. It needs about 80% to 98% of CPU usage. With much consideration given to maximizing and tweaking Web browser performance, you can find your favorite Web portals or magazines.

Client resources do not directly impact Web server performance too much. However, the way in which you obtain information from the Web servers will be impacted. Different resources available in the client generate different results. Different Web browsers also contribute.

Chapter 5. Security implications and performance

Security has become an extremely important issue, especially for Web-based scenarios that deploy with incredible speed. Some of the technologies have been developed for several years and are defacto standards for the next generation of business. Implementing a good security policy depends on a good understanding of all of its components and careful deployment. It also depends on how fast our applications will run after implementing security. Plus, it depends on knowing exactly our performance limitations and knowing our capacity frontier. You should be aware of all of these issues, especially for performing accurate capacity planning for current use or future growth.

This chapter examines the main security features available for Web serving inside an AS/400 system. This overview covers a brief introduction of some of these technologies. We encourage you to review all of the additional materials referred to in the sections that follow.

5.1 Security characteristics and components

The AS/400 system has strong system security characteristics. Consider these examples:

- AS/400 integrated security is extremely difficult to circumvent compared to security offerings on other systems that are add-on software packages.
- AS/400 object-based architecture makes it technically difficult to create and propagate a virus. On an AS/400 system, a file cannot pretend to be a program, nor can a program modify another program. AS/400 system integrity features require you to use system-provided interfaces to access objects. You cannot access an object directly by its address in the system. You cannot take an offset and turn it into or "manufacture" a pointer (*pointer manipulation* is a popular technique for hackers on other system architectures). However, viruses can actually be stored in the IFS. Therefore, they can potentially be passed on to other clients in the network, but the AS/400 system itself cannot be "infected" by it.
- AS/400 flexibility allows you set up your system security to meet your requirements. We strongly suggest that the minimum security level configured on an AS/400 system, which is supposed to act inside a scenario involving the Internet, should be at security level 30.
- The AS/400 system provides several security offerings to enhance your system security when you connect to the Internet. Virtual Private Networks (VPN), Network Address Translation (NAT), Digital Certificate Manager (SSL 3.0), and Firewall are features that are partly available on V4R1M0 of OS/400. However, they are fully available through Licensed Programs products on V4R4M0.

Before discussing these services, we first need to talk about the basic components of Internet security. These components include:

- A **security policy**: Defines what you want to protect and what you expect from your system users. It provides a basis for security planning when you design new applications or expand your current network. It should also include performance considerations. It describes user responsibilities, such as

protecting confidential information and creating nontrivial passwords. You need to create and enact a security policy for your organization that minimizes the risks to your internal network. The inherent security features of the AS/400 system, when properly configured, provide you with the ability to minimize many risks. However, it should have an impact on the overall performance of the system. When you connect your AS/400 system to the Internet, you need to provide additional security measures to ensure the safety of your internal network. If possible, you need to reduce the impact performance. Nevertheless, this cannot always be achieved. Performance and security are not related at all when you create your security policy. This is because if you need to setup security level 50 on your AS/400 system, there is nothing you can do in terms of improving performance or saving resources.

- **User authentication**: Ensures that only authorized individuals (or jobs) can enter your system. When you link your system to a public network like the Internet, user authentication takes on new dimensions. An important difference between the Internet and your intranet is your ability to trust the identity of a user who signs on. Consequently, you should seriously consider using stronger authentication methods than what traditional user name and password logon procedures provide. Digital certificates provide a stronger alternative while providing other security benefits. You should use digital certificates and SSL wisely to prevent unnecessary workloads on your system. Each time a user requests a certificate or a secure page, additional processing has to be done to complete the request.
- **Resource protection**: Ensures that only authorized users can access objects on the system. The ability to secure all types of system resources is an AS/400 strength. You should carefully define the different categories of users that can access your system. Also, you should define the access that you want to give these groups of users as part of creating your security policy.

It is critical that you understand the risks that are imposed by each service that you intend to use or provide. In addition, understanding possible security risks helps you to determine a clear set of security objectives. Once you understand the risks, you must ensure that your security policy provides the means for minimizing those risks. It also requires you to fully understand the different levels of security and how they may impact performance. For example, it is highly expected that you set the AS/400 security level to the level required after the creation of the security policy. However, this security level does not necessarily mean that you have to enable all of your HTML objects with SSL, especially if the pages you want to serve are static pages that do not contain sensitive data. Then, you do not need to enable SSL features for them. If they include any Server Side Includes, such as servlets, CGI-BIN, or Net.Data, which most of the time require some type of validation or access to several AS/400 objects, you should enable SSL for these types of dynamic pages.

Table 9 shows the correlation between the security policy and the technology that is available.

Table 9. Correlation between security policy and available technology

	Confidentiality	Integrity	Accountability	Autenthicity	Availability	Access Control	Auditability
SSL v3	Y	Y	IDENTITY	Y	N	MAPPINGS	LOGGING
Firewall	IPSEC	IPSEC	N	N	Y	Y	Y
Certification Authority	N	N	N	ID	N	N	N
SET	Y	Y	Y	Y	N	N	Y
Web Server	SSL	SSL	N	Y	N	Y	Y
Browser	SSL	SSL	N	N	N	N	N
VPN (IPSec)	Y	Y	N	SYSTEM	N	N	N
Anti-Virus	N	VIRUSES	N	N	Y	N	N
Network Sniffer	N	N	N	N	Y	N	Y
Java	SECURITY API	VERIFIER	N	N	Y	Y	N
Cookies	N	N	N	WEBSERVER	N	N	N

For detailed information on AS/400 security, please refer to the following documents:

- *AS/400 Tips & Tools for Securing Your AS/400*, SC41-5300
- *AS/400 Security - Basic V4R1*, SC41-5301
- *OS/400 Security - Reference*, SC41-5302
- *AS/400 Security - Enabling for C2*, SC41-5303

When you create and carry out a security policy, you must have clear objectives. You also want to be aware of which URLs need additional security due to their content.

For details on security policies, please refer to:

- *Internet Security in the Network Computing Framework*, SG24-5220
- *AS/400 Internet Security: Protecting Your AS/400 from HARM in the Internet*, SG24-4929
- *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162

5.2 Secure Sockets Layer

SSL is an acronym that stands for Socket Secure Layer. Figure 50 illustrates how SSL acts on the application layer.

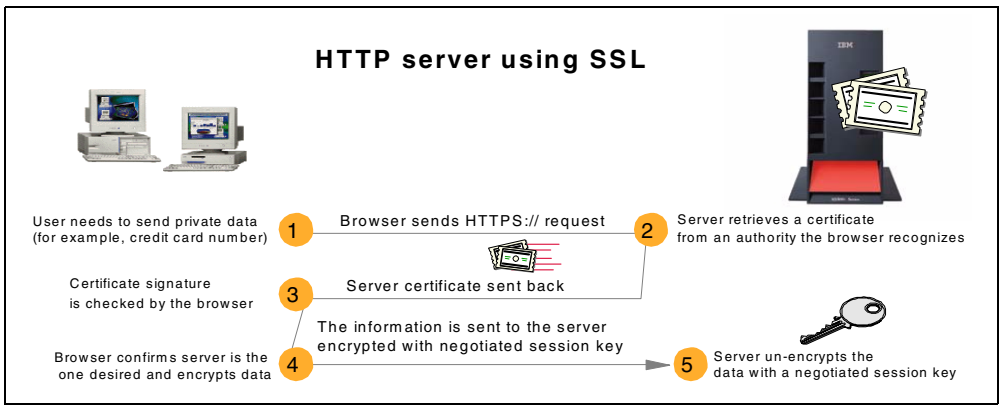


Figure 50. SSL process

SSL is a security protocol that was developed by Netscape Communications Corporation, along with RSA Data Security. It provides a private communications channel between the client and server for ensuring data encryption (privacy of data), authentication of session partners (digital certificates), and message integrity (digital signature). In theory, it is possible to run any TCP/IP application in a secure way without changing it. In practice, SSL is implemented for HTTP connections and some other applications, such as Client Access Express. During its securing process, for example, SSL has to encrypt and decrypt data. Therefore, it has a performance and resource impact. Figure 51 shows how an SSL handshake occurs.

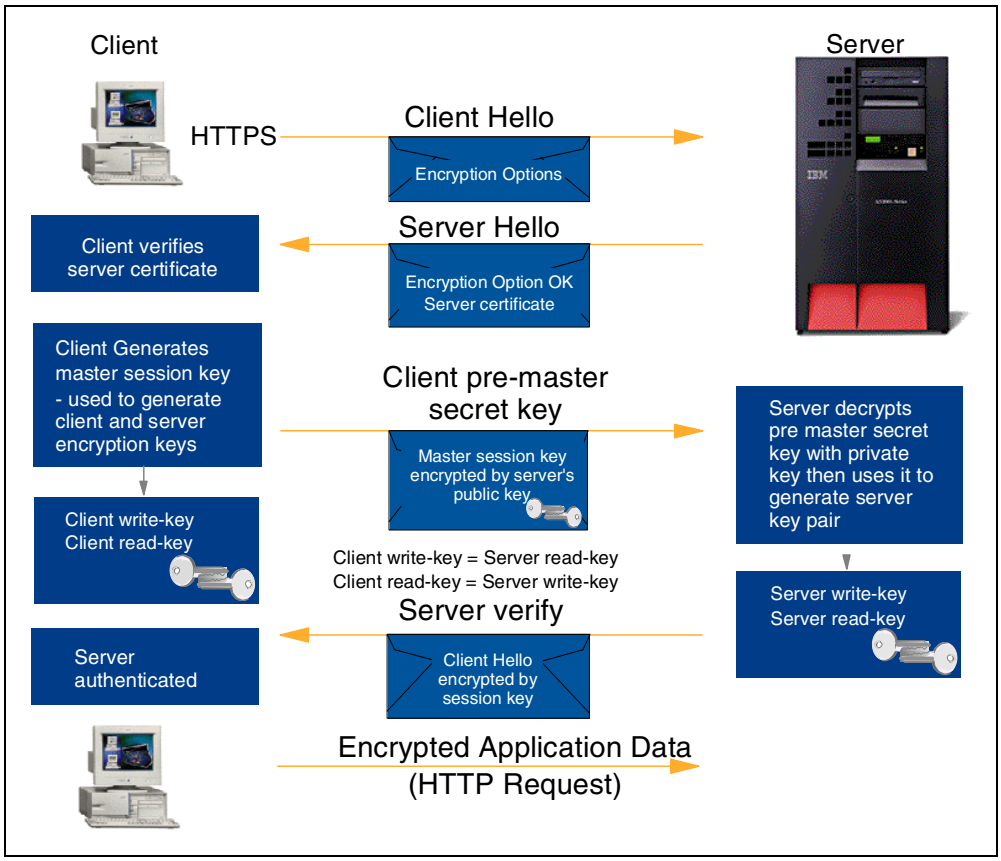


Figure 51. SSL handshake

When a client makes a secure connection with SSL for the first time, additional handshakes and processing must occur. This is referred to as a *full SSL handshake*. Once this is done, and the client's information can stay in the server's session cache, then *regular SSL handshakes* occur. The full SSL handshake can consume up to 20 times more CPU than the regular SSL handshake. In addition, depending on the kind of encryption you are using (either 128 bits or 40 bits), the performance impact may vary. Client authentication requested by the server is quite expensive in terms of CPU utilization and should only be used when needed.

5.2.1 SSL components

If SSL client authentication is configured, the server requests the client's certificate for any HTTPS (HTTP + SSL) request. The server establishes a secure

session, depending on whether the client has a valid certificate. This depends on the server configuration, such as no client authentication, optional client authentication, and mandatory client authentication. Once your server has a digital certificate, SSL-enabled browsers can communicate securely with your server using SSL. With SSL, you can easily establish a security-enabled Web site on the Internet or on your corporate network. SSL uses a security handshake to initiate the secure TCP/IP connection between the client and the server. During the handshake, the client and server agree on the security keys that they will use for the session and the algorithms they will use for encryption and to compute message digest or hashes. The client authenticates the server. In addition, if the client requests a document protected by SSL client authentication, the server requests the client's certificate. After the handshake, SSL is used to encrypt and decrypt all information on both the HTTPS requests and the server response, including:

- The URL that the client is requesting
- The contents of any form being submitted
- Access authorization information, such as user names and passwords
- All data sent between the client and the server

The benefits of HTTP using SSL include:

- The target server is verified for authenticity.
- Information is encrypted for privacy.
- Data is checked for transmission integrity.

Service	*Protection From*	*Technology*
Mutual Authentication	**Impostors**	X.509 Certificates
Message Integrity	**Vandals**	Message Authentication Codes (Keyed hash functions)
Message Privacy	**Eavesdroppers**	Encryption

Figure 52. Secure Sockets Layer provides three security services

HTTPS is a unique protocol that combines SSL and HTTP. You need to specify `https://` as an anchor in HTML documents that link to SSL-protected documents. A client user can open a URL by specifying `https://` to request SSL-protected documents. Because HTTPS and HTTP are different protocols and usually use different ports (443 and 80, respectively), you can run both secure and non-secure servers at the same time. As a result, you can choose to provide information to all users using no security, and specific information only to browsers that make secure requests. This is how a retail company on the Internet can allow users to look through merchandise without security, complete order forms, and send their credit card numbers using SSL security. A browser that does not support HTTP over SSL naturally cannot request URLs using HTTPS. The non-SSL browsers do not allow users to send forms that need to be submitted securely.

Figure 53. Accessing a secure session versus accessing a traditional session

In Figure 53, accessing a secure session versus accessing a traditional session requires additional work to be performed from both the client and the server.

5.2.2 SSL performance implications

The previous section explained that granting SSL on your Web site impacts server and client side performance. This is especially true because an extra job has to be performed by the server to encrypt and then decrypt the data. The greatest amount of time in an SSL session establishment is spent on the client (20 to 40% range of total time). The client has a lot of work to do. Client authentication requested by the server is quite expensive in terms of CPU and should only be requested when needed.

There are two major encryption methods: *symmetric (single key)* and *asymmetric (two keys)*. Symmetric encryption is much faster than asymmetric, but performance is not the only reason for choosing an encryption method. The major challenge with symmetric key encryption is key distribution. Since there is only one key, everyone must have a copy of this key. Asymmetric encryption uses two keys: a public key and a private key. The public key is freely distributed, while the private key is safe.

Session establishment delay is measurable and affects the end-to-end budget. The time it takes to set up an SSL connection should be considered seriously in the design. Hardware encryption can greatly reduce the response time and CPU time required for the SSL handshake. In conclusion, only use SSL when it is required.

5.3 Virtual Private Networks (VPNs)

Generally, the use of a VPN increases security. From a performance perspective, the use of a VPN impacts performance on the system by adding workload to the AS/400 resources. These resources, such as the communications IOP, have to perform extra processing from the traditional one (if you are serving Web pages). Every VPN that is configured generates a performance degradation, while the VPN session lasts. This degradation can generally be seen as an increase on the TCP/IP jobs running on the AS/400 system. It also includes the degradation of throughput from the Communications IOP or additional DASD space.

Additionally, VPN uses *strong cryptography*, since cryptography needs to codify the data workload increases. In V4R4, although AS/400 VPN does not use digital signatures and certificates for authentication, Digital Certificate Manager (5769-SS1 option 34) must be installed because there are several APIs provided by DCM that AS/400 VPN requires. VPN also requires Cryptographic Access Provider 5769-AC2 or AC3.

5.3.1 Objects used by VPN inside the AS/400 system

This section gives an overview of the objects that are created by a user or created by using a VPN.

5.3.1.1 VPN server jobs

VPN server jobs must be started before VPN connections can be initiated. VPN server jobs run in the QSYSWRK subsystem. These jobs are:

- **QTOKVPNIKE**: This is the Virtual Private Networking key manager job. The VPN key manager listens to UDP port 500 to perform Internet Key Exchange (IKE) protocols.
- **QTOVMAN**: This is the VPN connection manager job.

5.3.1.2 VPN policy database

Once you create your VPN, the associated configuration objects are stored in the VPN policy database. The VPN policy database consists of the objects in QUSRSYS as shown in Table 10.

Table 10. VPN generates these objects on the QUSRSYS library

Object	Type	Library	Attribute
QATOVDAH	*FILE	QUSRSYS	PF
QATOVDCDEF	*FILE	QUSRSYS	PF
QATOVDDFLT	*FILE	QUSRSYS	PF
QATOVDDSEL	*FILE	QUSRSYS	PF
QATOVDESP	*FILE	QUSRSYS	PF
QATOVDIID	*FILE	QUSRSYS	PF
QATOVDIPAD	*FILE	QUSRSYS	PF
QATOVDLID	*FILE	QUSRSYS	PF
QATOVDMCOL	*FILE	QUSRSYS	PF
QATOVDNATP	*FILE	QUSRSYS	PF
QATOVDN1	*FILE	QUSRSYS	PF
QATOVDPKEY	*FILE	QUSRSYS	PF
QATOVDRGRP	*FILE	QUSRSYS	PF
QATOVDR1	*FILE	QUSRSYS	PF
QATOVDSRVR	*FILE	QUSRSYS	PF
QATOVDUCP	*FILE	QUSRSYS	PF
QATOVD1PRP	*FILE	QUSRSYS	PF
QATOVD1SP	*FILE	QUSRSYS	PF
QATOVD1TRN	*FILE	QUSRSYS	PF
QATOVD2LST	*FILE	QUSRSYS	PF
QATOVD2PRP	*FILE	QUSRSYS	PF
QATOVD2SP	*FILE	QUSRSYS	PF
QATOVD2TRN	*FILE	QUSRSYS	PF
QTOVDVPKEY	*VLDL	QUSRSYS	
QTOVDVSKEY	*VLDL	QUSRSYS	
QTOVDBJRN	*JRN	QUSRSYS	

Review the following related documents for more information:

- *A Comprehensive Guide Virtual Private Networks, Volume I: IBM Firewall, Server Client Solutions*, SG24-5201
- *A Comprehensive Guide Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234
- *A Comprehensive Guide Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309
- *IBM Firewall for AS/400 V4R3: VPN and NAT Support*, SG24-5376
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- Internet Engineering Task Force (IETF) Request for Comments:
  - Security Architecture for the Internet Protocol
  - IP Authentication Header (AH)
  - The Use of HMAC-MD5-96 within ESP and AH

5.3.2 VPN performance implications

VPN works at the IP layer, rather than at the socket layer as with SSL. Therefore, it is typically used to secure a broader class of data, rather than SSL. All of the data flows between two systems rather than, for example, just the data between two applications. Securing a connection using VPN is completely transparent to the application.

The performance of VPN varies 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. While VPN-ESP can perform authentication, AH-only affects system performance half as much as 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.

5.4 Firewall

Because a firewall represents a substantial portion of your network security policy, we encourage you to understand exactly what a firewall is and what a firewall can do for you. The performance implications of using a firewall are specific to your firewall vendor. However, we can give you some recommendations on the IPCS Firewall for AS/400.

Using the Integrated PC Server as a firewall provides additional value-add for the AS/400 system as a Web server. The firewall can be on the same system as the Web server or on a different system within the network. With the Integrated PC Server handling the firewall activity, the AS/400 CPU is not significantly impacted.

If a system is not optimally configured, the decrease can be more significant. For example, if the MTU size is reduced to 500 bytes, the impact of the firewall can be a 50% capacity reduction.

In a scenario where the Web server is behind the firewall, the Integrated PC Server performs packet filtering and allows HTTP traffic only through to the Web server (also on the same AS/400 system). For an optimally configured system, having the firewall function active under a load only slightly degrades the overall AS/400 Web server capacity (compared with a similar, non-firewall configuration).

5.5 Internet security terminology

To establish a basis for discussing Internet security, we define some Internet terms. If you already know these terms, you may continue on to Chapter 6, “Sizing Web-based applications” on page 91.

Algorithm
: The computational procedures used to encrypt and decrypt information.

Cryptography
: The science of keeping data secure. Cryptography allows you to store information, or to communicate with other parties, while preventing

non-involved parties from understanding the stored information or understanding the communication. *Encryption* transforms understandable text into an unintelligible piece of data (*cipher text*). *Decryption* restores understandable text from unintelligible data. Both processes involve a mathematical formula or algorithm and a secret sequence of data (the key).

There are two types of cryptography:

- In *shared/secret key (symmetric) cryptography*, one key is a shared secret between two communicating parties. Encryption and decryption both use the same key.
- In *public key (asymmetric) cryptography*, encryption and decryption each use different keys. A party has two keys: a public key and a private key. The two keys are mathematically related, but it is virtually impossible to derive the private key from the public key. A message that is encrypted with someone's public key can be decrypted only with the associated private key. Alternately, a server or user can use a private key to "sign" a document and use a public key to decrypt a digital signature. This verifies the document's source.

Decryption

The process of reverting encrypted information (cipher text) back to plain text.

Digital certificate

A digital document that validates the identity of the certificate's owner, much as a passport does. A trusted party, called a *certificate authority* (CA) issues digital certificates to users and servers. The trust in the CA is the foundation of trust in the certificate as a valid credential. Use them for:

- ***Identification***: Knowing who the user is.
- ***Authentication***: Ensuring that the user is who they say they are.
- ***Integrity***: Determining whether the contents of a document have been altered by verifying the sender's digital "signature".
- ***Non-repudiation***: Guaranteeing that a user cannot claim to not have performed some action. For example, the user cannot dispute that they authorized an electronic purchase with a credit card.

For more information, refer to the redbook *Safe Surfing: How to Build a Secure WWW Connection*, SG24-4564.

Digital signature

On an electronic document, this is equivalent to a personal signature on a written document. A digital signature provides proof of the document's origin. The certificate owner "signs" a document by using the private key that is associated with the certificate. The recipient of the document uses the corresponding public key to decrypt the signature, which verifies the sender as the source.

Digital Certificate Manager (DCM)

Registers certificates that you create on your AS/400 system when it is acting as a certificate authority. You can also use the DCM to register certificates that other certificate authorities issue. DCM allows you to

choose to associate a user's certificate with their AS/400 user profile. You can use DCM to associate digital certificates with various AS/400 applications so that these applications can use the Secure Sockets Layer for secure communications.

Distinguished name
The name of the person or server to whom a certificate authority issues a digital certificate. The certificate provides this name to indicate certificate ownership. Depending on the policy of the CA that issues a certificate, the distinguished name can include other information.

Encryption
Transforms data into a form that is unreadable by anyone who does not have the correct decryption method. Unauthorized parties can still intercept the information. However, without the correct decryption method, the information is incomprehensible.

Extranet A private business network of several cooperating organizations located outside the corporate firewall. An extranet service uses the existing Internet infrastructure, including standard servers, e-mail clients, and Web browsers. This makes an extranet more economical than creating and maintaining a proprietary network. It enables trading partners, suppliers, and customers with common interests to use the extended Internet to form tight business relations and a strong communication bond.

Intranet An organization's internal network that uses Internet tools, such as a Web browser or file transfer protocol (FTP).

IPSec A set of protocols to support the secure exchange of packets at the IP layer. IPSec is a set of standards that AS/400 and many other systems use to carry out Virtual Private Networks (VPNs).

Key A value that causes a cryptographic algorithm to run in a specific way and to produce a specific cipher text (for example, a 128-bit key).

Network Address Translation (NAT)
Provides a more transparent alternative to the proxy and SOCKS servers. It also simplifies network configuration by enabling networks with incompatible addressing structures to be connected. NAT provides two major functions. It can protect a public Web server that you want to operate from within your internal network. NAT provides this protection by allowing you to hide your server's "true" address behind an address that you make available to the public. It also provides a mechanism for internal users to access the Internet while hiding the private internal IP addresses. NAT provides protection when you allow internal users to access Internet services because you can hide their private addresses.

Secure Sockets Layer (SSL)
Originally created by Netscape, SSL is the industry standard for session encryption between clients and servers. SSL uses symmetric key encryption to encrypt the session between a server and a client (user). The client and server negotiate the session key during an exchange of digital certificates. The key expires automatically after 24 hours. A different key is created for each client and server connection. Consequently, even if unauthorized users intercept and decrypt a

session key (which is unlikely), they cannot use it to eavesdrop on later sessions.

Virtual Private Network (VPN)

An extension of an enterprise's private intranet. You can use it across a public network such as the Internet, creating a secure private connection, essentially through a private "tunnel." VPNs securely convey information across the Internet, connecting other users to your system. These include:

- Remote users
- Branch offices
- Business partners or suppliers

For additional information on these topics, please refer to the following list of publications:

- *A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions*, SG24-5201
- *Safe Surfing: How to Build a Secure WWW Connection*, SG24-4564
- *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162
- *AS/400 Internet Security: Securing your AS/400 from HARM in the Internet*, SG24-4929
- *IBM Firewall for AS/400 V4R3: VPN and NAT Support*, SG24-5376
- *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162
- *An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, Communications, and PC Connectivity*, GG24-4200
- *HTTP Server for AS/400 Web Masters Guide*, GC41-5434

Chapter 6. Sizing Web-based applications

This chapter discusses sizing the necessary resources, including client, server, and network. The primary emphasis is on AS/400 server resources. This chapter also goes through a number of practical examples.

6.1 Sizing basics

Sizing can be broadly defined as the process of determining the appropriate resource requirements, given a particular set of inputs. Generally, sizing is done when you have new projects or significant changes from status quo, as opposed to capacity planning, in which you have historic data and plan to do forecasts.

Sizing Web-based applications takes on two types of analyses. First, you can determine the client, server, and network resources necessary, given an expected Web application load (which can be comprised of static pages and application generated pages). The second analysis treats the resources available as a set of "fixed" resources and the application load as the dependent variables.

6.2 Sizing for different application characteristics

In Chapter 4, “Correlation of measured performance data” on page 65, we discuss the characteristics of Web applications, including static pages, dynamically generated pages, persistence and caching, and communications infrastructure environments. This section focuses on categorizing a Web application workload. For specific product characteristics, refer to Appendix A, “IBM application framework for e-business” on page 187.

6.2.1 Static Web page serving

Static Web pages are existing documents (text, tags for graphic elements, and other objects such as Java applets) that reside on your AS/400 server, generally in the AS/400 Integrated File System (IFS), but possibly also in the QSYS library file system. These pages can be retrieved from disks or loaded in the local cache during server startup. The standard metric of hits per second per CPW gives a good basis for sizing the server load. However, you must ensure that you factor in the AS/400 operating system version and page size. These are from various versions of the AS/400 Performance Capabilities Reference documents. Refer to Table 11 for static Web page sizing guidelines.

Table 11. Static Web page sizing guidelines

AS/400 release	V4R2	V4R3	V4R4
Hits/second/CPW	.66	1.00	1.18
Hits/second/CPW - cached	NA	1.25	1.86
Hits/second/CPW, secure (40-bit RC4)	.29	.44	.48
Hits/second/CPW, secure (40-bit RC4) - cached	NA	.54	.58

Note that substantial improvements are inherent in newer versions of the AS/400 operating system. The impact of using secure sockets for encryption and authentication is sizeable.

6.2.2 Static Web page serving example

Let's perform a moderately simple sizing for a proposed Web site. This site will contain static pages, initially. The following input accurately represents the Web pages and site:

- 80% of the pages will be, on average, 1 KB of text and an additional three graphic elements of 10 KB each.
- The remaining 20% of the pages will be, on average, 5 KB of text and an additional five graphic elements of 10 KB each.
- The AS/400 server is a Model 170, with feature code 2291, and is on V4R4 or V4R3.
- We will assess the load at 100,000 hits evenly spread over 24 hours.
- We will assess the impact of 50% of the objects being accessed from a local cache, instead of a disk.

Here are the calculations for the un-cached scenario and V4R4:

```
small pages: 100k*80%/(24hrs/*3600)*[1 + 3(1.08)] = 3.9 'hits/second'
large pages: 100k*20%/(24hrs/*3600)*[1.04 + 5(1.08)] = 1.5 'hits/second'
utilization: 5.4 hits/second / (115 CPW * 1.18 hits/second/CPW) = 4%
general formula (v4r4) : utilization = 4.6 / server CPW
general formula (v4r3) : utilization = 5.4 / server CPW
```

This gives us a "load" of 5.4 hits per second. If we have a Model 170, with feature code 2291 and a processor CPW value of 115, this represents about 4% on average. Of course, peak or maximum load time uplift factors will increase this percentage. Also, if the AS/400 server was at V4R3, the utilization would be about 5%.

If we factor in caching, the V4R4 metric changes from 1.18 to 1.86 hits/second/CPW. If we achieve 50% of our "hits" in cache, we can use the average of cached and non-cached (1.52) values to recalculate our utilization. In this example, the utilization drops to 3%. If we apply the same rationale to a V4R3 server, the hits/second/CPW is 1.12 and the utilization is about 4%.

6.2.3 Dynamic Web page serving and application sizing

Many of the documents served from Web servers in Internet and intranet type environments are static Web pages. However, you eventually need to extend your Web presence and include dynamic pages. Examples of dynamic pages include search engines, requests for information, account inquiries, and ordering merchandise on line. There are a variety of dynamic Web page techniques ranging from including JavaScript in your HTML documents, through advanced technologies such as Java servlets and server pages.

Sizing dynamically-generated Web pages is much more complex than sizing static pages. The HTTP server must still analyze the browser request and deliver the HTML document as it does with static requests. However, the server will likely do additional application processing, such as retrieving account information

stored in a database. Additionally, since the pages are dynamically created, you may not have as much opportunity to decrease your server load via local caching.

The tables in various versions of the *AS/400 Performance Capabilities Reference*, SC41-0607, list the relative hits per second per CPW metric for a variety of dynamic Web page environments. This document is available on the Web at: `http://as400bks.rochester.ibm.com/pubs/html/as400/onlinelib.htm`

The server processing is significantly more than with static Web pages. For example, Table 12 shows the performance planning metrics for Net.Data in two environments: with and without an SQL database operation.

Table 12. Net.Data performance metrics

AS/400 Release	V4R2	V4R3	V4R4
Hits/second/CPW - no SQL	.09	.14	.24
Hits/second/CPW - no SQL, secure (40 bit RC4)	.07	.12	.19
Hits/second/CPW - with SQL	.05	.11	.15
Hits/second/CPW - with SQL, secure (40 bit RC4)	.04	.10	.13

Note that V4R3 provides almost twice the improvement over V4R2, and V4R4 provides almost three times the improvement over V4R2. The "with SQL" environment involves a fairly simple database retrieval. Certainly, more sophisticated server functionality is available and deployed on the AS/400 system. Obviously, the more server application processing that takes place, the more server processing resources are consumed.

Outside of elementary database query-oriented applications, you need to determine how much additional impact is involved in the server tasks for your application.

6.2.3.1 Example application sizing

In Chapter 3, “Performance measurement and analysis tools” on page 27, we discuss measurements for the Web application environment. An important tool for measuring the server load component for dynamic Web application environments, such as CGI, Net.Data, or Java servlets, is the AS/400 Performance Monitor transaction trace. If you deploy your own applications, they may have characteristics that are different from those used to build the hits/second/CPW tables referenced earlier. For example, you may have a complicated database query component to your application that needs to have appropriate uplift factors for any sizing exercises.

Figure 54 on page 94 shows an example of a transition report obtained from our AS/400 performance monitor trace.

```
 Job type  . . : BD     Elapsed Time -- Seconds          Sync/Async Phy I/O   -MPL-
                        ----------------------          ------------------------ C   I     Last 4 Programs in
Invocation Stack
             State Wait   Long   Active   Inel    CPU    DB    DB  NDB   NDB      u   n
-----------------------------------------
    Time     W A I Code   Wait    /Rsp*   Wait    Sec    Read  Wrt Read  Wrt  Tot r   l    Last     Second     Third
Fourth
 ------------ ----- ---- ------- ------- ------- -------- ---- ---- ---- ---- ---- -- -- ---------- ----------
---------- ---------
 13.35.34.879 *TRACE ON
 13.35.48.576 ->A        13.697                                                  9
 13.35.48.612 W<-                  .036            .025         2         1      9    QSOSRV1    QZHBHJOB   QZHBHJOB
QZHBHJOB
                                                        1                      PAG=    0           PWrt=    4
  ----------  QZHBHJOB            .036*          .025   0   2   0   1   3*     PAG=    0           PWrt=    4
                                                      0   1   0   0   1
 13.36.16.836 ->A        28.224   .002           .002                      10
 13.36.16.871 W<-                 .036           .025        2        1     10
                                                        1                      PAG=    0           PWrt=    4
  ----------  QZHBHJOB            .038*          .027   0   2   0   1   3*     PAG=    0           PWrt=    4
```

Figure 54. AS/400 performance transition report example

This report is for a CGI program that has database read and write functions. It is only for one job, so you would have to add the other job threads associated with the transaction. Now, let's look at a section of the transition report (Figure 55) when there were static Web page requests.

```
Job type  . . : B      Elapsed Time -- Seconds          Sync/Async Phy I/O   -MPL-
                       ----------------------          ------------------------ C   I    Last 4 Programs in Invocation
Stack
             State Wait   Long   Active   Inel    CPU    DB    DB  NDB   NDB      u   n
-----------------------------------------
    Time     W A I Code   Wait    /Rsp*   Wait    Sec    Read  Wrt Read  Wrt  Tot r   l    Last     Second     Third
Fourth
------------ ----- ---- ------- ------- ------- -------- ---- ---- ---- ---- ---- -- -- ---------- ----------
---------- ---------
13.38.07.540 ->A          .004                                                   7
13.38.09.710 W<-                  2.170            .005             4           10    QSOSRV1    QZHBSUTL   QZHBSUTL
QZHBSUTL
 ----------  QZHBSUTL           2.170*          .005   0   0   4   0   4*
13.38.11.558 ->A         1.847                                                   7
13.38.11.558 W<-                   .001                                              6    QZHBHTTP   QZHBHTTP   QZHBSUTL
QP0WPTHR
 ----------  QZHBSUTL            .001*                0   0   0   0   0*
13.38.28.494 ->A        16.936    .001           .001                       9
13.38.28.495 W<-                                                          9
 ----------  QZHBSUTL            .001*          .001   0   0   0   0   0*
13.38.28.503 ->A          .009                                                   9
13.38.30.680 W<-                  2.177            .003    1        3           4    QSOSRV1    QZHBSUTL   QZHBSUTL
QZHBSUTL
```

Figure 55. AS/400 performance transition report showing static page requests

Figure 55 shows a simple example since we are looking only at the main job thread for the HTTP server job. However, it shows that CPU time is consumed and is substantially less than that of the CGI job.

For sizing your Web-based applications, you may need to perform one or more AS/400 Performance Monitor transaction traces to estimate the relative CPU consumption for your particular situation.

6.2.3.2 Dynamic Web page serving example

Let's perform another moderately simple sizing for an enhancement to the Web site listed in 6.2.2, "Static Web page serving example" on page 92. We still have our static Web page workload calculated earlier. Now we want to embellish the site by enabling customers to peruse a customized product description and price list. We use Net.Data to analyze the user's request, query the products and prices database, and build an HTML response page. The following inputs accurately represent the Web pages and site:

- 30% of the Web site visitors will request a customized product and price list.
- The total server application consumes about twice the CPU utilization of the SQL task (in other words, twice the uplift of a Net.Data SQL page compared to an HTML page).
- The output pages will be, on average, 10 KB of text and an additional five graphic elements of 10 KB each.
- The AS/400 server will be on V4R4 or V4R3.
- The site traffic is 100,000 hits evenly spread over 24 hours.
- We want to enable secure pages between the user and the HTTP server.

Since this is an addition to our existing Web site, we must calculate the incremental load of the dynamic application and add it to our existing load for serving static pages. Let's determine the appropriate hits/second/CPW metrics first.

For V4R4, our hits/second/CPW ration for secure SQL-based Net.Data to secure HTML-based Net.Data is .19/.13 (about a 50% uplift). Our performance monitoring tests show that we should double this uplift to account for additional complexity and processing. We determine that we should plan for .09 hits/second/CPW (.19/(1+50%*2)) for V4R4.

For V4R3, our secure SQL Net.Data to HTML Net.Data ratio is .12/.10 (20% uplift). Again, we double this uplift factor to account for application complexity. Our factor calculates to .09 hits/second/CPW as it did for V4R4 (.12/(1+20%*2)).

We must also calculate the impact of the pages being 10 KB in size, plus the graphic elements and their addition to the encryption overhead. For the base page, our uplift factor is 8% for a 10 KB page. For the graphic elements, we use this same uplift factor. We also use the appropriate static page hits/second/CPW (.58 for V4R4, and .54 for V4R3 for secure pages).

```
base pages: 100k*30%/(24hrs/3600)*[1.08] = .4 'hits/second'
base page utilization: .4 hits/second /(115 CPW * .09 hits/second/CPW) = 4%
graphics: 100k*30%/(24hrs/3600)*[5*1.08] = 1.9 'hits/second'
graphics utilization: 1.9 hits/second / (115 CPW * .58 hits/second/CPW) = 3% (V4R4)
graphics utilization: 1.9 hits/second / (115 CPW * .54 hits/second/CPW) = 3% (V4R3)
total incremental utilization = 7%
general formula: utilization = 7.8 / server CPW
```

This gives us a load of 2.3 hits per second, represented by dynamic and static page requests. If we have a Model 170, with feature code 2291 and a processor CPW of 115, this represents about 7%, on average. Peak or maximum load time uplift factors will increase this percentage.

In our static page example, without caching, our CPU utilization for V4R4 and V4R3 was about 4% and 5% respectively. Adding in the dynamic content increases this considerably to 11% and 12% respectively. Some of the increase is due to encrypting the graphic elements. Some is due to the additional server processing for the Net.Data application.

6.3 Sizing AS/400 resources

General AS/400 resource sizing involves CPU, disk capacity and number of arms, main memory, and communications IOP. Since the server processes many jobs simultaneously, individual response times will vary based on the resource utilizations. If you look at overall HTTP server performance and response time, the actual time is comprised of the resource service time and the resource waiting or queue time. Because of this complexity and number of independent variables, you need to correlate overall response time with resource utilization on the AS/400 system. For example, suppose you measured processor utilization over a period of time for a constant workload of 20 transactions per second, and determined it was 10% of the system CPU. We may run the same task, generating a workload of 50 transactions per second and find that it consumes 40% system CPU. The increased load gives a lower average time (1/50 seconds instead of 1/20 seconds per transaction), but at the expense of a significantly higher resource utilization.

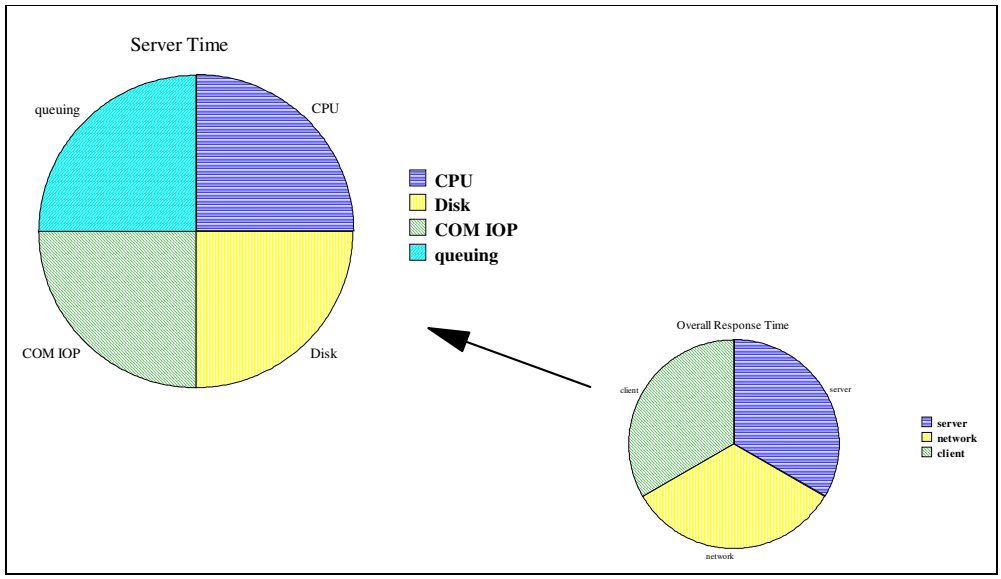


Figure 56. Server considerations in Web application performance

Queuing is a factor for each resource. It increases quickly as the resource utilization increases.

6.3.1 Main processor

In 6.2, “Sizing for different application characteristics” on page 91, we discuss a number of techniques for estimating the processor load for a given Web application environment. Generally, the AS/400 system will run multiple applications. These workloads need to be added to the HTTP serving requirements. There are no hard and fast rules for sizing these "other" applications. Every customer environment will be different.

AS/400 workloads are typically a combination of batch jobs (such as HTTP serving) and interactive jobs (such as an interactive order entry application). You should consider AS/400 server models or 7xx models rather than standard system models such as a Model 640. These give higher capacity for batch type jobs. Additionally, some of the older server models actually experience a

reduction in batch capacity if the interactive capacity reaches a particular threshold. Consider this simple example:

- HTTP server and associated applications requirements: 300 CPW
- Other batch and non-interactive workload requirements: 300 CPW
- Interactive workload requirements: 50 CPW

We can solve this workload with a Model 170 server, with feature code 2388, assuming that an interactive workload is not expected to grow much (maximum interactive CPW is 82, but anything over 70 will take batch processing resources). We can also solve this with a Model 720-2063, with feature code 1502. This would support a more interactive workload, but a smaller batch workload.

In general, a one-way CPU machine should not exceed 80% utilization. N-way machine utilization should not exceed 90%.

6.3.2 Disk arms

Planning the disk storage amount requirements for the AS/400 server is not difficult. However, it involves much more than simply identifying the amount of gigabytes required. From a cost perspective, you might assume that using the largest disk drives possible (such as 17 GB drives currently available) is most advantageous. From a server perspective, often times the opposite is true, depending on the workload. The AS/400 server requires enough disk drives (referred to as disk arms) to yield optimum performance from the AS/400 processors.

Fortunately, there are good sizing recommendations available from the AS/400 home page on the Internet at:
`http://www.as400.ibm.com/developer/performance/as4armct.html`

We approach planning for disk arms from these perspectives:

- AS/400 system and server considerations
- Light, mixed, or heavy disk workload environments
- Using the disk arm recommendations tables and formulas
- General DASD performance guidelines

6.3.2.1 AS/400 system and server considerations

AS/400 system models generally run a significant amount of on-line transaction processing workloads. These are recommended for heavy use of interactive, 5250-based applications. These applications are typically heavy disk users, with a significant amount being random, record-oriented retrieval and update tasks. Although AS/400 server models are recommended for HTTP serving, you may have situations in which a system model is acting as a Web server.

AS/400 server models (and the 7xx series) are specially tuned for batch and client/server application environments. These are generally recommended for HTTP serving, Lotus Domino, data warehousing and mining applications, and other tasks not running as 5250 interactive jobs. These applications can have drastically different characteristics. Simple static Web page serving, especially those with heavy graphical content, places relatively small, sequential workloads on disk. Dynamic application models, such as CGI, Net.Data, or Java servlets, may place much heavier demands on disk resources, especially if you are doing complex queries or other database activities.

6.3.2.2 Disk workload environments for AS/400 server models

Categorizing your disk workload can be a challenge, particularly since the AS/400 machine is running a variety of workloads and applications. You need to make some assumptions based on the three categories of the disk arm recommendation charts: light, heavy, and mixed disk workload environments.

The light disk workload recommendations suggest a minimum number of disk arms necessary for acceptable performance. Serving static Web pages fits this model. The analysis that you need depends on the AS/400 machine, feature, and appropriate controller.

The heavy disk workload is for transaction-oriented applications running in a "client/server" mode. Dynamic Web pages, particularly those doing database tasks such as ordering products or account inquiries, fit this category.

The mixed disk workload environment is essentially a 50/50 mix of the light and heavy disk workload environments. This is likely the most realistic environment. However, the 50/50 mix is merely a guideline. You need to determine the split that makes sense for your environment.

6.3.2.3 Using the disk arm recommendation tables and formulas

To do your own sizing, you may want to print the AS/400 Disk Arm Requirements Based on Processor Model Performance document at the Web site:
`http://www.as400.ibm.com/developer/performance/as4armct.html`

6.3.2.4 General DASD performance guidelines

You should refer to the *AS/400 Performance Capabilities Reference* manual for your particular operating system release and AS/400 disk IOP and arms. We generally recommend that you do not exceed a utilization of 60% for your disk IOPs and 50% for your disk arms. You can use the AS/400 Work with Disk Status (`WRKDSKSTS`) command to get a quick view of your disk status. An example is shown in Figure 57.

```
                          Work with Disk Status                    AS20
                                                          08/26/99  09:18:36
Elapsed time:   00:00:07

             Size    %     I/O   Request   Read  Write   Read  Write    %
Unit  Type    (M)  Used    Rqs  Size (K)    Rqs   Rqs     (K)   (K)   Busy
   1  6713   8589  82.2     .4       4.0     .0    .4      .0   4.0      0
   2  6713   6442  82.2     .5       5.0     .0    .5      .0   5.0      0
   3  6713   6442  82.2     .8       4.0     .1    .6    4.0    4.0      0
   4  6713   6442  82.2     .5       4.0     .1    .4    4.0    4.0      0
   5  6713   8589  82.2     .5       4.0     .0    .5      .0   4.0      0
   8  6713   6442  82.2     .1       4.0     .0    .1      .0   4.0      0
```

Figure 57. Example WRKDSKSTS display

6.3.3 Main memory

AS/400 main memory is analogous to Random Access Memory (RAM) on workstations, such as personal computers. In either case, your system will run on a minimum amount of memory. However, the swapping between chip memory and DASD can become unbearable. You can add memory, but you will eventually

reach a point where adding memory provides minimal returns on your incremental investment.

The AS/400 server assigns main memory to storage pools. Pools can be shared, which means that jobs from multiple subsystems can be run. Four storage pools are defined by default:

- *MACHINE for AS/400 system jobs
- *BASE, which contains the minimum value specified by the QBASPOOL system value plus any unassigned main memory
- *INTERACT for interactive jobs
- *SPOOL for print jobs

You can also create private storage pools in which memory is allocated to a single subsystem's jobs. This subsystem can use the private storage pool, as well as the shared pools.

The AS/400 server's single level storage and other inherent system administration features make it a system that can manage itself. If a particular job requests data that is not already in the storage pool, it must retrieve it from disk. This is called a *page fault*. As requested data is retrieved, other data must be recorded in auxiliary storage, which is called *paging*. Our main memory and storage pools need to be sized appropriately to keep paging down to a reasonable level.

6.3.3.1 Impact of page faults on performance

A page fault adds 10 to 30 milliseconds to end user response time, based on the time to read data from disk into main memory. If a transaction has 5 faults, it adds .05 to .15 seconds to the end user response time.

As transactions increase, a given fault level has less effect. If you have an average of 10 faults/second and 100 transactions/second, only about 10% of the transactions will be affected. For such batch jobs as the HTTP server, a guide of 12 faults/second is acceptable. The percentage of time spent in the page fault would be:

```
fault % = faults/second * disk response time * 100
```

For a batch job with one transaction/second, 12 faults/second, and assuming a 10 millisecond disk response time, 12% of the time spent on page faults. If you have 100 transactions per second, on average, 12% of the transactions will incur paging, which means 1.2% of the time spent on page faults. If our faults/second increased to 100 (the maximum if using automatic performance adjustment), your 100 transactions per second metric increases to 10% of the time spent in paging.

In addition to an increase in overall response time, page faulting consumes some cycles of the AS/400 processor and increases disk input/output operations.

6.3.3.2 Main memory sizing considerations

Refer to *AS/400 Work Management V4R4*, SC41-5306, for your particular operating system level. This section summarizes a few key points relative to the HTTP server application.

The AS/400 *MACHINE storage pool will need a certain amount of memory, depending on the total memory on the system, the number of system jobs, number of communications lines and protocols, system functions deployed such as Double-byte character set, and the amount of disk arms. You may also need to allocate a certain amount to the *QINTER pool. Follow this guide:

```
*QINTER pool = (main storage - *MACHINE pool - *QSPL pool) * .7
```

The *QSPL pool size depends on the number of active writers. This pool size will likely be 1.5 to 2 MB.

You need to factor in all expected workloads to determine how much to make available to the *BASE and any private storage pools. For batch jobs, we recommend a range from about 1.5 MB for short running jobs, to 16 MB for compiled programs. A good standard is to allow 2 MB per batch job.

For HTTP serving, the minimum recommended amount of main memory is 1 MB + (.5 MB* number of HTTP server instances). In addition, you need to factor in the main memory necessary for any application processing. For example, you may be using Net.Data to query the database for an order inquiry application. You may want to allocate additional memory specific to this application:

```
2 MB for HTTP server + 2 MB * (10 database jobs) = 22 MB
```

The HTTP server for releases V4R3 and later enable caching. This allows the HTTP server to reduce the overhead of retrieving a specific Web object from disk. You can use HTTP server caching specifically, or the AS/400 Set Object Access (SETOBJACC) command to load your Web page objects into memory. If using caching, you need to provide enough main memory so that the objects remain in main memory. Otherwise, the AS/400 memory management process may swap these objects out of main memory. Caching may be of some benefit, since it reduces the overhead associated with a file open from disk.

Since most AS/400 implementations run mixed workloads, it is often not worth the additional system administration investment that is necessary for setting up private storage pools, overusing the system defaults, and letting the AS/400 operating system automatically adjust the *MACHINE pool (based on the QPFRADJ system value). If your operating system is at V4R3 or later, the AS/400 HTTP server runs in its own subsystem, QHTTPSVR. If you want to deploy a dedicated HTTP server at V4R3 or later, you may want to create a private storage pool for the QHTTPSVR subsystem. This reserves main memory specifically for the HTTP server jobs. Plus, memory from the *BASE storage pool can also be utilized.

Planning for memory requirements and the associated performance tuning can be quite complicated. In most cases, you may find it most advantageous to use the AS/400 shared storage pools only, rather than trying to configure and manage private pools.

6.3.4 Communications IOPs

The AS/400 server uses Input Output Processor (IOP) cards to minimize the amount of resource that the main CPU processors need to expend for peripheral operations, such as disk access and communications. IOP cards contain memory, processor, and input/output capabilities. From an application serving point of view, the communications IOP introduces some latency to the response time

(processing and queue time). This is a small price to pay since it enables our main processor to do more application work, including database operations.

For sizing communications resources, it is extremely important to assess local area network (LAN) and wide area network (WAN) requirements. In our own private network, we typically have LAN-based hosts accessing our server. Typical HTTP application requests involve small sends by the client and larger sends to large responses from the server. The clients may have equal or faster network interface cards than our server. Network bandwidth can be a serious constraint and may require multiple LAN cards in our server or special network hardware such as LAN switches to enable full duplex communications. We cover this in more detail in 6.6, “Capacity and response time sizing: Network” on page 110.

The communications IOP requirements are influenced by these factors:

- Network type
- Bandwidth
- Other limiting factors

The current network infrastructure may dictate the amount of flexibility that is available. Having Ethernet and category 5 cabling deployed is much better than trying to run TCP over twinaxial connections on the AS/400 system. Similarly, a WAN with Synchronous Optical Network (SONET) or Asynchronous Transfer Mode (ATM) is advantageous over dial-up phone lines or X.25.

Bandwidth is another key factor. If the AS/400 system is performing Internet serving and you have a T1/E1 line to your Internet Service Provider (ISP), you can get by with one LAN IOP in a routed network, from a bandwidth perspective (T1 is 1.5 Mbps and E1 is 2 Mbps, obviously much less than even a 4 Mbps Token-Ring connection). In a LAN environment, a general rule for the Web application environment is that active end users need 50 Kbps, and that at any given time, 5% to 10% of the end users must be active.

Outside of bandwidth limitations, there are AS/400 LAN IOPs that have a limit of transactions per second. This is approximately 120 transactions per second if you are using Ethernet or Token Ring IOP cards. In V4R4, Ethernet IOPs can be dedicated to TCP only and support approximately 240 transactions per second. Redundancy is another important factor. You may have a high enough volume of network workload that warrants multiple or redundant network links. Another consideration is that it may be impractical or infeasible to put more than a certain amount of IOPs in a particular AS/400 system.

6.3.4.1 Sizing examples: LAN IOP

Let’s estimate the communications IOP requirements for our earlier example in 6.2, “Sizing for different application characteristics” on page 91. The requirements are:

```
100k hits/day * 80% 'small pages' * (1k + 10k*3) = 28.7 KBs
100k hits/day * 20% 'large pages' * (5k + 10k*5) = 12.7 KBs
100k hits/day * 30% 'secure pages' * (10k + 5*10k) = 20.8 KBs
```

Total average bandwidth required = `62.2 KBs`

Certainly, you need to uplift this number to cover peak load situations. However, it appears easily containable with one LAN IOP. You should calculate the number of hits per second to ensure that you do not exceed 120.

```
100k hits/day * (80% * 4 + 20% * 6 + 30% * 6) /(24 * 3600) = 7.2 hits/second
```

You are well within the limits for a single communications IOP.

If you have an intranet Web application environment and substantial LAN traffic, you may have a very heavy "desired" bandwidth, but a more modest "can do" capacity.

Let's look at another example with more robust communications requirements. Let's say during a given work day, you have:

- 1000 "intranet" application users accessing your AS/400 system
- 500 interactive (5250) sessions accessing your AS/400 system
- 1000 Lotus Notes users using e-mail and applications
- 500 users using other client/server applications

You already sized the intranet application requirements based on handling all demands. However, you may not be able to afford unlimited network capacity and need to account for the other applications that may be running. The necessary bandwidth and activity level per user is certainly open to debate. However, consider these guidelines:

- Browser application users need 50 Kbps, and 10% will be active at any given time.
- 5250 application users need 20 Kbps, and 30% will be active at any given time.
- Lotus Notes users need 40 Kbps, and 30% will be active at any given time.
- Other client server applications need 40 Kbps, and 20% will be active at any given time.

First, complete the math for the bandwidth requirements as shown here:

```
Intranet:        1000 users * 10% active * 50 Kbps = 5 Mbps
Interactive:     500 users * 30% active * 20 Kbps = 3 Mbps
Notes: 1000      users * 30% active * 40 Kbps = 12 Mbps
Client/server:   1000 users * 20% active * 40 Kbps = 8 Mbps
Total:           28 Mbps
```

From a bandwidth perspective, you generally do not want to exceed 50% of the aggregate bandwidth for Token-Ring networks or 30% for Ethernet. You should also look at the transaction rate before making a decision on the number of LAN IOPs for your AS/400 system:

```
Intranet:        1000 users * 10% active * 1 hit/15 seconds * 5 objects/hit
Interactive:     1000 users * 5% active * 1 transaction / 10 seconds
Notes:           1000 users * 10% active * 1 transaction / 15 seconds
Client/server:   1000 users * 10% active * 1 transaction / 10 seconds
Total:           55 transactions per second
```

We made some rough estimates on the number of transactions per second per user, as well as the number of objects per page. Certainly, you need to make estimates that reflect your environment.

Based on these estimates, you should plan on three 16 Mbps Token-Ring IOPs, eight to ten 10 Mbps Ethernet IOPs, or one 100 Mbps Ethernet IOP. If you are

using multiple IOPs in your AS/400 system, you need to configure your TCP parameters to support load balancing.

```
                          Display TCP/IP Route
                                                              System:   MYAS400
Route destination  . . . . . . . . . . :   *DFTROUTE
Subnet mask  . . . . . . . . . . . . . :   *NONE
Type of service  . . . . . . . . . . . :   *NORMAL
Next hop . . . . . . . . . . . . . . . :   10.11.12.1
Preferred binding interface  . . . . . :   10.11.12.25
Maximum transmission unit  . . . . . . :   *IFC
Duplicate route priority . . . . . . . :   5
Route metric . . . . . . . . . . . . . :   1
Route redistribution . . . . . . . . . :   *NO
```

Figure 58. TCP route configuration

This particular AS/400 server has several LAN IOPs that are all connected to the network backbone. We can create multiple default routes, one for each TCP interface. If we assign the same duplicate route priority across multiple LAN interfaces, we distribute the load across the communication IOPs. In this example, we bind this particular TCP/IP interface on a specific IOP to a specific router segment on our network.

6.3.4.2 Sizing example: WAN IOP

If you take the calculations from the first example in the previous section (62 Kbps bandwidth and 7 hits/second) and add 50% to the requirements to provide for a peak demand scenario, you must plan for a bandwidth of 93 Kbps. You can solve this in one of three ways:

- Two WAN IOPs supporting three V.34 modems, which produces a total bandwidth of approximately 100 Kbps (probably not a realistic solution)
- One WAN IOP with an ISDN Terminal Adapter attached, which means 128 Kbps of total bandwidth
- One WAN IOP that supports two 56 Kbps Frame Relay, serial, or X.25 lines for a total usable bandwidth of 112 Kbps

6.4 HTTP server attributes

There are a number of characteristics pertaining to our AS/400 HTTP server that have a modest to substantial impact on server load and end user response time. In some cases, such as the AS/400 file system being used, there is measurable data to assist us with our sizing exercise. In other cases, there is no data and you may have to experiment or develop your own.

6.4.1 AS/400 file system considerations

Up to this point, we have not mentioned anything about performance impacts based on the AS/400 file system being, for example, root, QSYSLIB, QOPENSYS, or QDLS. Does this have an effect?

The answer is yes, a substantial effect. This best performance is obtained from the root and QOPENSYS file systems, as shown in Table 13.

Table 13. File system performance impact

File system	Hits/sec/CPW factor
root	1
QOPENSYS	1
QDLS	.8
QSYS	.4
QLANSRV	.2
QNETWARE	.2

6.4.2 Caching

The IBM HTTP server for AS/400 has supported local caching since V4R3. As seen in the hits/second/CPW metrics, it can make a sizeable difference (for V4R4 cached pages, it is 1.86 hits/second/CPW, but non cached is only 1.18). You may not realize that level of difference in your environment. With local caching, your Web server is basically pre-approved for "file open" tasks when the server is started. The actual objects may be on disk or in main storage. Certainly, a read from main memory is much faster than from disk. However, you have no guarantees where the data resides, unless you have an extremely large amount of memory on your server.

With caching, a hash table is built on the AS/400 system listing all cached objects. For each HTTP object request, this table is consulted to determine if any of the requested objects are cached. If you compare caching to non-caching, cached pages take less CPU and should be retrieved substantially faster. Non-cached pages will incur a slight performance hit because the cache hash table must be consulted.

Caching works best and is easiest to implement if it is applied to the most frequently accessed objects, such as the HTML pages and selected graphics files in a particular directory on our server. Also, once a cached object is changed, it is removed from the cache table. Objects that are subject to editing or changes are not good cache candidates.

V4R4 introduces a concept called *dynamic caching*. This can provide greater flexibility than static caching because the most frequently accessed objects can be cached, rather than having to statically specify the objects when you start the server. This tends to work best for small Web sites with a limited amount of objects that will be potentially served. However, on larger sites the dynamic caching algorithm may use more server resources to manage the cache than what is saved by retrieving objects in the cache.

From a sizing perspective, caching can reduce a server load substantially in some cases. The key criteria is that certain HTML files and graphics files are accessed much more frequently than others and these should be cached. If most objects on your site are accessed with equal likelihood, caching will be of little benefit and can actually be detrimental. If you have certain heavily accessed pages, then caching should be helpful. However, you need to ensure that your

server has adequate main memory (and low page fault rates). Also, your cached content must be relatively stable. You may want to be conservative in doing your sizing analyses. For example, you may want to figure V4R4 static pages with caching to be 1.5 hits/second/CPW (the average of cached and non cached).

6.4.3 Persistent connections

The AS/400 HTTP server supports persistent connections in V4R3 and later releases. However, you must be cautious in using these connections, because there are a number of scenarios that may negatively impact performance, for example:

- The browser may be slow in closing the request, which leaves the connection open and the server thread unavailable for other uses.
- The server keeps the request active until the persistence timeout is reached if the maximum requests per connection has not been realized.
- If persistence values are set too high, and all the server threads are occupied, new client requests will time-out, which has a major negative impact.

Persistent connections are most beneficial in a LAN intranet-oriented environment where response time is generally less than the persistence timeout. In fact, many commercial Internet sites turn off persistent HTTP connections completely. From an AS/400 perspective, you may want to set the maximum requests per connection parameter to 1 to avoid problems for Internet users.

6.4.4 HTTP server directives

There are numerous capabilities in the AS/400 HTTP server that can yield important functional features. Keep in mind that many come with a slight performance penalty. There are also a few values you can enter to improve overall performance:

- UseACLs (default is protectonly; do not set at always)
- UseMetaFiles (default is off)
- MinActiveThreads (try to set at, or close to, the MaxActiveThreads)
- MultiFormatProcessing (default is all; set to none by using the AS/400 Work with HTTP Configuration (`WRKHTTPCFG`) command)
- DNS-Lookup (default is off)
- InputTimeout (default is 2 minutes; you may want to reduce this)
- proxy (default is off)
- no_proxy (default is none if proxy being used; set this for your trusted intranet addresses and hosts)
- AccessLog, AgentLog, etc. (logging takes some system resource)
- AccessLogExcludeHostName, AccessLogExcludeMimeType, etc. (only log what you really need to)
- AddClient (default is none; use browser detection only if necessary)

Another trick you may want to use is to put your most used request statements (your map, pass, redirect and exec statements) ahead of lesser used statements in your server's configuration file. Each request to the HTTP server goes through

the request processing statements in the order listed in the configuration file until it finds a match or fails.

If you are hosting multiple Web sites, you may also want to consider whether you should have multiple HTTP server instances (one for each customer, for example) or combine the attributes for each one in the same HTTP configuration file. Here is an example of assigning multiple welcome pages:

```
Welcome welcomeibm.html www.myserver.com
Welcome welcome400.html www.my400server.com
```

In this case, the AS/400 HTTP server is servicing two Web sites (www.myserver.com and www.my400server.com), using different IP addresses, different ports, or the HTTP 1.1 virtual hosts feature available on V4R3 and later releases. This makes management and administration easy because you have just one configuration file and one server instance and set of jobs running. However, each request may require a slight amount of extra processing because the server must not only service each request, but it must also look at the host requested and determine if a special course of action is necessary, for example, based upon host www.myserver.com or www.my400server.com.

If your AS/400 system has suitable processor and memory resources, you may find it more advantageous to create a separate configuration file and server instance for each Web site. Certainly, this will consume more server resources, but may give better response time. You may have to experiment with both if you are hosting multiple sites on the same AS/400 system, since each situation will be different.

6.5 Response time sizing: Client

In many client/server application environments, the actual client application is often considered a negligible part of the overall response time. Certainly, workstation price/performance ratios continue to consistently improve. However, applications have also increased their demand of client processing resources for graphically-rich user interfaces. Plus, the HTTP communications connect/disconnect environment places demands on the user's workstations.

Since the end user's perspective of the Web site and Web applications performance is based on response time at the workstation and browser, it is important that you understand the impact at the client. Additionally, you need to establish a suitable overall response time baseline to factor in the network and server components.

6.5.1 Web browser and client performance considerations

The Web browser client is often overlooked when analyzing Web application performance. However, it contributes to overall response time. Client performance is another set of publications in itself. For our purposes, we will view it fairly simply (Figure 59).

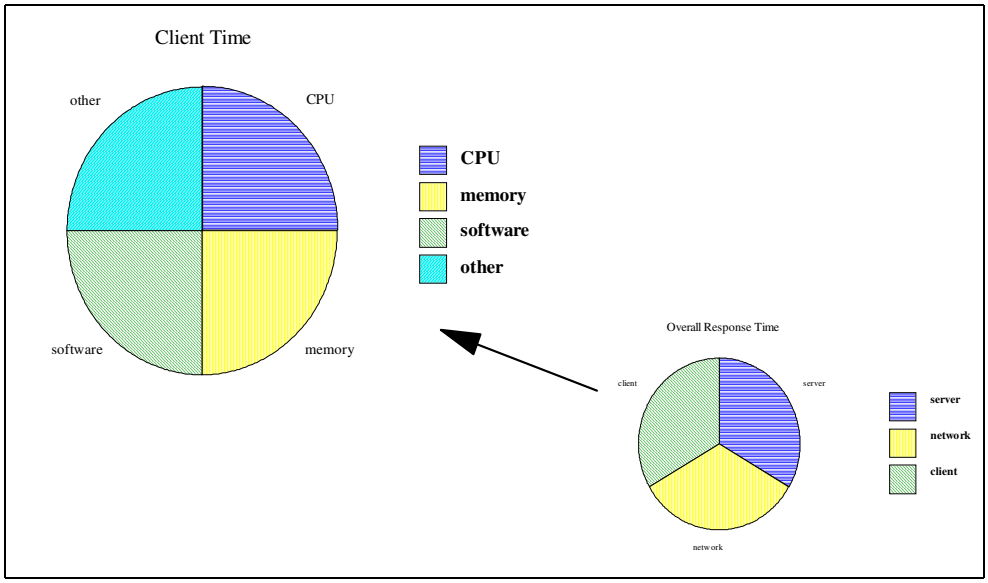


Figure 59. Client considerations in Web application performance

At a minimum, it takes a discernible amount of time for an HTML page to be rendered at the end user's browser. Those in leading edge information technology roles, such as application developers and system administrators, typically have new, high-powered workstations with plenty of RAM and CPU, plus the latest operating system and fixes. However, many of your customers may not have this level of workstation, and the response time at the browser can be significant.

6.5.1.1 Web page response time

Web page response time is the elapsed time between a page request and when it is rendered in the browser. As mentioned earlier, this is comprised of contributions from the client, network, and server. Measuring response time can be simple (time it with a stop watch) or complex (Internet-based third parties offer Web response time analysis reports, for a fee). Commercial applications, such as the Tivoli Application Response Measurement API, can also be used, or you can build our own by using a few simple JavaScript functions in your Web page. Figure 60 on page 108 shows an example of how you can "instrument" your Web page.

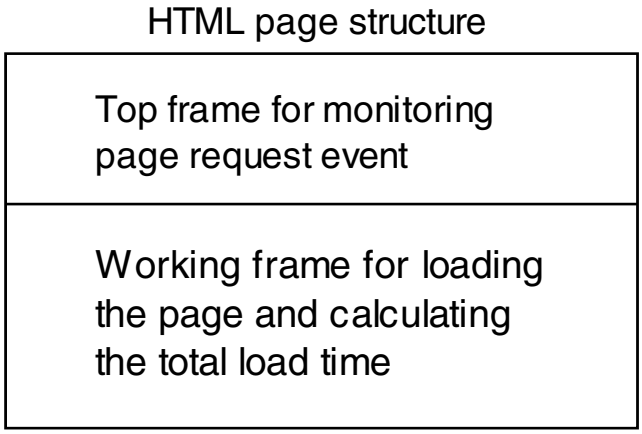


Figure 60. Web page response time measurement

The HTML source for the top frame is fairly simple (Figure 61). It provides a means for entering a Web page address and timestamping when the request was made.

```
<html>
<head><title>Browser Response Time Test</title>
<script>
function loadnewframe(){
clicktime = new Date()
document.enterurl.clicktimet.value = parseInt(clicktime.getTime())
top.frames[1].location = urlstring
}</script>
</head>
<body >
<h3>Use this page to tabulate browser response times</h3>
<p>Start URL (also try newtest.html, newtesta.html or newtestb.html):
<form method="post" name="enterurl" >
<input type="text" size="30" name="startingurl" value="dd_mail_lab.html">
<input type="button" name="newurl" value="go to url"
onClick="loadnewframe()">
<input type="hidden" name="clicktimet" >
</form>
</body>
</html>
```

Figure 61. Top frame HTML source

The user enters an appropriate URL or file and clicks on the Go to URL button. This triggers the loadnewframe function, which sets the current time and converts it into milliseconds. Then, it loads the URL or file in the bottom frame.

The HTML source for the bottom frame has additional code (Figure 62). It contains JavaScript that sets two timestamps: one when the head portion of the page loads and another after the entire page loads. Browsers execute all HTML tags in the head section, including scripts, prior to processing the HTML tags in the body section.

```
<html>
<head><title>Test Page</title>
<script>
function workwindowstart() {
startup = new Date() }
function workwindowload() {
loadbody = new Date()}
workwindowstart()
function calcloadtime(){
workwindowload()
pageload = parseFloat(parseInt(startup.getTime() -
parent.topf.document.enterurl.clicktimet.value)/1000)
loadtime = parseFloat(parseInt(loadbody.getTime() -
startup.getTime())/1000)
documentload = parseFloat(parseInt(loadbody.getTime() -
parent.topf.document.enterurl.clicktimet.value)/1000)
alert("base page load time (seconds) = "+pageload+"\ndocument load time =
(seconds) ="+loadtime
+"\ntotal download time (seconds) = "+parseFloat(pageload+loadtime))}
</script>
</head>
<body onLoad="calcloadtime()">
<p>hello world
<img src="aclock1.gif"><br>
<img src="hot-apr98.gif">
<img src="beanwiz01.gif">
</body>
```

Figure 62. Working frame HTML source

The page shown in Figure 62 processes the scripting functions prior to loading the rest of the page elements. The workwindowstart function adds a timestamp when the head section is loaded. When the entire document body is loaded, the browser's onLoad event occurs. This triggers the calcloadtime function, which sets another timestamp. Next, we use the timestamp from the top frame as our time basis (the hidden variable clicktimet in form enterurl, which is in frame topf of our parent HTML document). The pageload variable measures the time, in milliseconds, for the head section of the document to be loaded. The loadtime variable measures the time, in milliseconds, between the head section load and the entire document load. The total download time is then the sum of pageload and loadtime. The results then appear on a popup window as shown in Figure 63.

Figure 63. Example of Web page load time using JavaScript

6.5.1.2 Measuring the client contribution

As part of this redbook, we tested the client impact on overall Web page response time. To isolate the workstation's contribution to total response time, we tested loading several identical Web pages from the local workstation, using the same browser type and version. The intent was to enable a simple client response formula such as this (where `c` is a constant, `x` and `y` are variables):

```
client response time = c + # of objects (x) + amount of data (y)
```

We used two workstations: a laptop PC (133 MHZ processor, 72 MB RAM, and Windows 98) and a desktop PC (350 MHZ processor, 256 MB RAM, and Windows NT 4). The response time readings were obtained by using JavaScript functions in the HTML page to monitor the total time from the "page click" until the browser document's page load completed.

Table 14. Time to load local Web pages

Scenario	1 object, .8 KB data	4 objects, 86 KB data	4 objects, 109 KB data	37 objects, 569 KB data
Laptop PC - Response time	.28 sec	1.24 sec	2.34 sec	9.97 sec
Desktop PC - Response time	.04 sec	.16 sec	.3 sec	2.64 sec

Note that the desktop PC response time is much faster. If we go through the necessary math for a linear regression goodness of fit test, we get these approximations for predicting the client's time to load a page:

```
Laptop PC: time = .14 - .02(# of objects) + .02(KB of data)
Desktop PC: time = .04 + .09(# of objects) - .001(KB of data)
```

Obviously we must exercise some caution in using these formulas. Intuition tells us that response time should be a positive constant, and another constant multiplied by the number of objects, plus still another constant multiplied by the amount of data. The key is to measure response time for pages that accurately reflect your Web site and use the formulas as a guide. However, this can be useful in determining the browser client's contribution to total response time, in addition to factoring in the network and server components.

Table 15. Predicted Web page loading time: Client contribution

Scenario	5 objects, 50 KB data	10 objects, 100 KB data	15 objects, 200 KB data	30 objects, 500 KB data
Laptop PC	.97 sec	1.8 sec	3.55 sec	8.82 sec
Desktop PC	.43 sec	.81 sec	1.13 sec	2.09 sec

Our simple analysis does not factor in the page load time based on the workstation's CPU or memory utilization. In these examples, both were quite low. Also, HTML pages with a significant amount of JavaScript or a local Java applet must be factored in as well.

In summary, the client has a measurable impact on overall response time. Best performance is achieved by a combination of hardware (CPU, RAM, disk access, network card), software (operating system, browser), and some local tuning (disk defragmentation).

6.6 Capacity and response time sizing: Network

In the previous section, we discuss a fairly easy means for determining the client's contribution to overall response time. This section looks at the impact of the network and communications infrastructure. You may find that this component is much more complicated and difficult to analyze than the client. The variability is also much greater (especially if the Internet is involved). Figure 64 summarizes the network considerations.

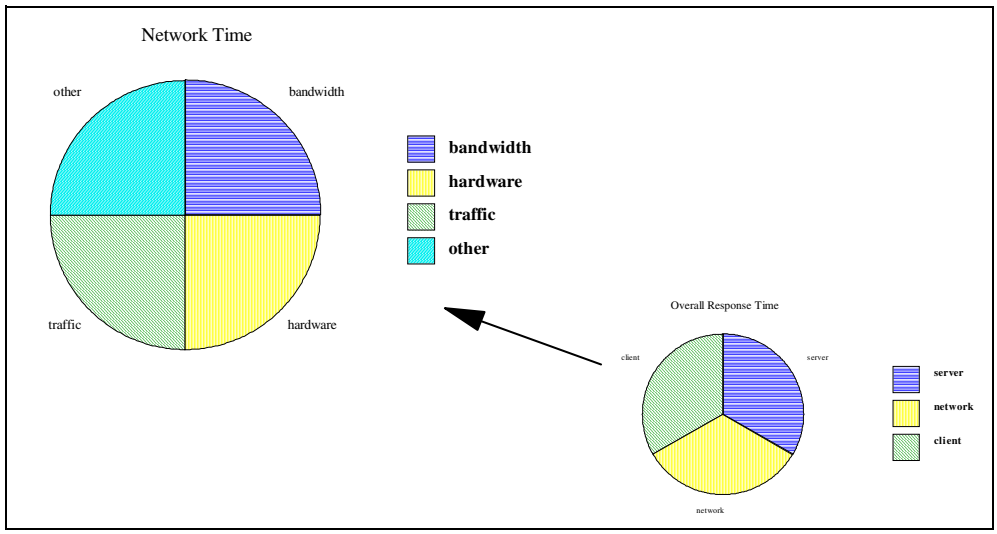


Figure 64. Network considerations in Web application performance

The network and communications infrastructure contribute a certain time element in addition to our client time. Factors include bandwidth (100 or 10 Mbps Ethernet, T1 wide area links, 28.8 Kbps modem), the network topology (routers, bridges, switches, and their firmware algorithms), firewalls and application proxies, circuit versus packet switched lines, utilization of individual network segments, quality of service deployments, and numerous other factors.

In a wide area environment, such as the Internet, or offices connected with T1/E1 or ISDN lines, bandwidth is significantly less and latency is higher. For example, you may deploy an ISDN router with 128 Kbps capability. Your network provider may have one or more T1 (1.5 Mbps) connections to an Internet point of presence location, which eventually links to the Internet. Each network segment and piece of hardware adds delays and even greater variability.

6.6.1 Network sizing considerations

In 6.3.4, “Communications IOPs” on page 100, we show a few simple examples on sizing network requirements. Since network bandwidth and infrastructure capacity comes at a price, affordability and feasibility often dictate over purely technical requirements. A switched network with all of our servers having full duplex 100 Mbps capability would be great, but cannot be implemented. Since most IP-based applications involve several send/receive sequences per transaction, response time and overall throughput are highly variable because each send or receive has variation.

To illustrate this complex subject, we again used the Web page response time analysis tool to determine the impact and variability of adding in the network component to overall response time. The network environment is a 16 Mbps Token-Ring. See Table 16.

Table 16. Network impact on overall response time

Scenario	86 KB page	109 KB page	139 KB page	569 KB page
Local laptop PC	1.15 seconds	2.3 seconds	2.75 seconds	10.22 seconds
Network laptop PC	1.38 seconds	2.47 seconds	2.91 seconds	18.59 seconds

Scenario	86 KB page	109 KB page	139 KB page	569 KB page
Uplift factor	1.2	1.07	1.06	1.82
Local desktop PC	.17 seconds	.34 seconds	.39 seconds	3.52 seconds
Network desktop PC	.38 seconds	.59 seconds	.69 seconds	3.98 seconds
Uplift factor	2.24	1.74	1.76	1.13

Note the tremendous variability in the results. In actuality, we factored in the server and network. We ran the tests when the server CPU utilization was less than 5% so its variability is modest. The response time and its variability introduced by the network is extensive. For example, the response time differences between connecting to the server on the same network and across networks connected by routers was negligible. In fact, many times the response time was better across the router hops. Our network time reflects the average of a one hop and three hop network.

If we calculate a simple arithmetic average, the uplift factor for the laptop PC is 1.29 over the static page load time. For the desktop PC, it is 1.7. Many commercially available products, such as the IBM NWays Manager or Cisco's CiscoWorks 2000 product, can be used to monitor network throughput, bandwidth availability, and utilization. To get an idea of the variability of network traffic, we used the What's Up product from IPSwitch which helped us to understand the network bandwidth available in our local network. As shown in the following example, the only traffic is between our PC tool simulating multiple Web users, and our AS/400 Token-Ring IOP:

```
Check 192.168.1.1 20/1000/6000/1000; Start time 08/30/99 15:39:50
Pkt:1    Sent:      56 BRec:      56 BTime:       2.06Throughput: 448.00 Kb/s
Pkt:2    Sent:     106 BRec:     106 BTime:       9.08Throughput: 188.44 Kb/s
Pkt:3    Sent:     156 BRec:     156 BTime:       2.09Throughput: 1.24 Mb/s
Pkt:4    Sent:     206 BRec:     206 BTime:       3.00Throughput: 1.09 Mb/s
Pkt:5    Sent:     256 BRec:     256 BTime:       3.06Throughput: 1.36 Mb/s
Pkt:6    Sent:     306 BRec:     306 BTime:      14.06Throughput: 349.71 Kb/s
Pkt:7    Sent:     356 BRec:     356 BTime:      29.01Throughput: 196.41 Kb/s
Pkt:8    Sent:     406 BRec:     406 BTime:       4.07Throughput: 1.62 Mb/s
Pkt:9    Sent:     456 BRec:     456 BTime:       3.06Throughput: 2.43 Mb/s
Pkt:10    Sent:     506 BRec:     506 BTime:       3.09Throughput: 2.69 Mb/s
Pkt:11    Sent:     556 BRec:     556 BTime:      14.05Throughput: 635.42 Kb/s
Pkt:12    Sent:     606 BRec:     606 BTime:       4.04Throughput: 2.42 Mb/s
Pkt:13    Sent:     656 BRec:     656 BTime:       4.05Throughput: 2.62 Mb/s
Pkt:14    Sent:     706 BRec:     706 BTime:       5.01Throughput: 2.25 Mb/s
Pkt:15    Sent:     756 BRec:     756 BTime:      33.09Throughput: 366.54 Kb/s
Pkt:16    Sent:     806 BRec:     806 BTime:       6.02Throughput: 2.14 Mb/s
Pkt:17    Sent:     856 BRec:     856 BTime:       5.02Throughput: 2.73 Mb/s
Pkt:18    Sent:     906 BRec:     906 BTime:      24.01Throughput: 604.00 Kb/s
Pkt:19    Sent:     956 BRec:     956 BTime:      24.03Throughput: 637.33 Kb/s
Pkt:20    Sent:    1000 BRec:    1000 BTime:       5.02Throughput: 3.20 Mb/s
End time 08/30/99 15:40:12
20 packets, 21228 bytes in 207 ms. average:820.40 Kb/s median:1.36 Mb/s
```

Even on a private network, the bandwidth varies immensely.

6.6.2 Data traffic considerations

In a pure Web serving and HTTP environment, client and server applications can tolerate periodic, unpredictable delays because of the connectionless mode of TCP and HTTP protocols. However, if you have Systems Network Architecture (SNA) traffic, there is less tolerance for unpredictable delays. Similarly, any NetBIOS Enhanced User Interface (NETBEUI) traffic can add a relatively high amount of overhead to the network, such as LAN broadcasts. Each protocol has

its own management and keep alive structure. Obviously, the more protocols that are run on a particular segment, the more overhead is involved. Similarly, connecting network segments that support multiple protocols influences our devices, such as routers, bridges, and hubs. Typically, devices dedicated to the IP only can operate much faster than those that have to support multiple protocols.

As mentioned earlier, somewhat unpredictable response times are particular to Web- and HTTP-oriented applications. Other applications, such as terminal emulation, are much more sensitive to delays. Since most installations running the IP protocols do not use any quality of service mechanism, you can have file transfers or large e-mail attachment downloads that wreak havoc on your network performance.

6.6.3 Other traffic considerations

Many organizations maintain separate networks for voice and data traffic. However, voice over IP and IP multicasting are bringing to the desktop new types of bandwidth-hungry applications such as audio and video. Some of these applications, such as telephone voice traffic, are extremely time delay sensitive. Others use streamed audio or video and heavy usage of buffers. For example, you can listen to live radio station broadcasts over the Internet with as little as 8 Kbps bandwidth available (with significant compression).

6.6.4 Quality of service

The IP protocol packet header has a type of service field, which can be used for specifying class or quality of service, similar to that of SNA. Numerous technologies and products based on Resource Reservation Protocol (RSVP) and Differentiated Services (Diff-Serv) have been introduced. The premise is that time sensitive data, such as audio and critical application traffic (such as a credit card order), should be prioritized higher than file transfers or downloading large e-mail attachments. Most products attempting to solve this problem rely upon vendor-specific solutions.

Another technology family attempting to solve this problem is Directory or Application Enabled Networking. This basically involves setting up priority policies in a Lightweight Directory Access Protocol (LDAP) enabled directory. A system administrator can then set application priorities and network hardware, such as routers, and switches can prioritize traffic based on these parameters. As with RSVP and Diff-Serv, most solutions are vendor-specific.

6.6.5 Redesign or more bandwidth

Usually the easiest solution, and least expensive from a short term basis, is more bandwidth. For example, replacing 10 Mbps Ethernet cards with 100 Mbps cards is fairly common. For Internet access and home or small office environments, technologies such as cable modems and various forms of Digital Subscriber Lines (DSL) can be attractive. However, such technologies as cable modems and forms of DSL have much higher download rates than upload rates. This is fine for the Web surfer but not for the server infrastructure that has to try filling these faster network pipes. Complexity and cost can quickly add up, especially if you have operations in multiple countries.

However, recent advances in networking hardware make it advantageous to selectively redesign our network for better performance and growth potential. LAN switches are an important factor in achieving a better network design, especially in a local or intranet environment. LAN switches originated as fast bridge type devices operating at the MAC layer of the communications protocol. Their advantage was enabling dedicated full bandwidth and an isolated collision domain on each port of the switch. This enabled a server's LAN connection to be run in full duplex mode, plus enabling workgroups to have their own dedicated bandwidth. Newer LAN switches can operate at the network layer and even the transport layer, which gives router software type functionality in a high performance hardware package.

Here is an example of a network redesign in a campus environment, taken from the IBM Networking Web page at:
`http://www.networking.ibm.com/case/studies.html`

In this example, the problem is poor performance between subnetworks and other routed portions of our network (Figure 65). The routers that were installed in 1994 have not kept up with the increased bandwidth requirements of the network. While the wiring closet shared media hubs were replaced with switches, which increased the bandwidth available to the applications, the routers that are the gateway to the FDDI backbone were not upgraded. The solution should address today's problem and position the customer for future network upgrades when they are required.

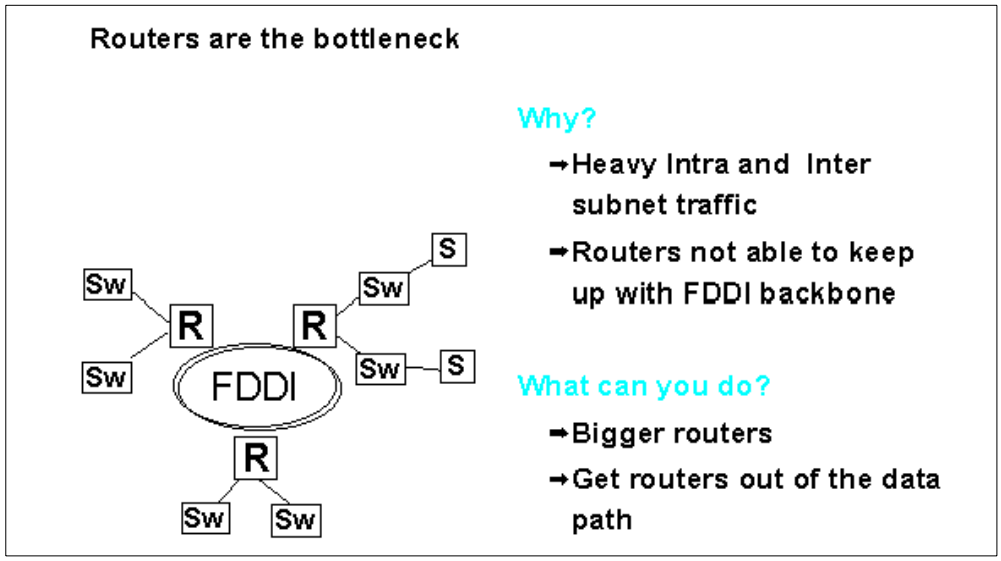


Figure 65. Poorly performing network design

A potential solution is to use bigger, more powerful routers with a higher packets per second forwarding capacity metric. This will certainly work, but may not be an optimal solution. Routers have a great deal of flexibility and may have specialized processors to maximize performance, but they still rely on firmware and software processing.

Switches, on the other hand, have higher packets per second forwarding metrics than routers. They are also more flexible than routers in that they allow you to

segment collision domains and provide full duplex communications, where appropriate.

In our example, we solve the problem by replacing routers with layer 3 network switches (Figure 66).

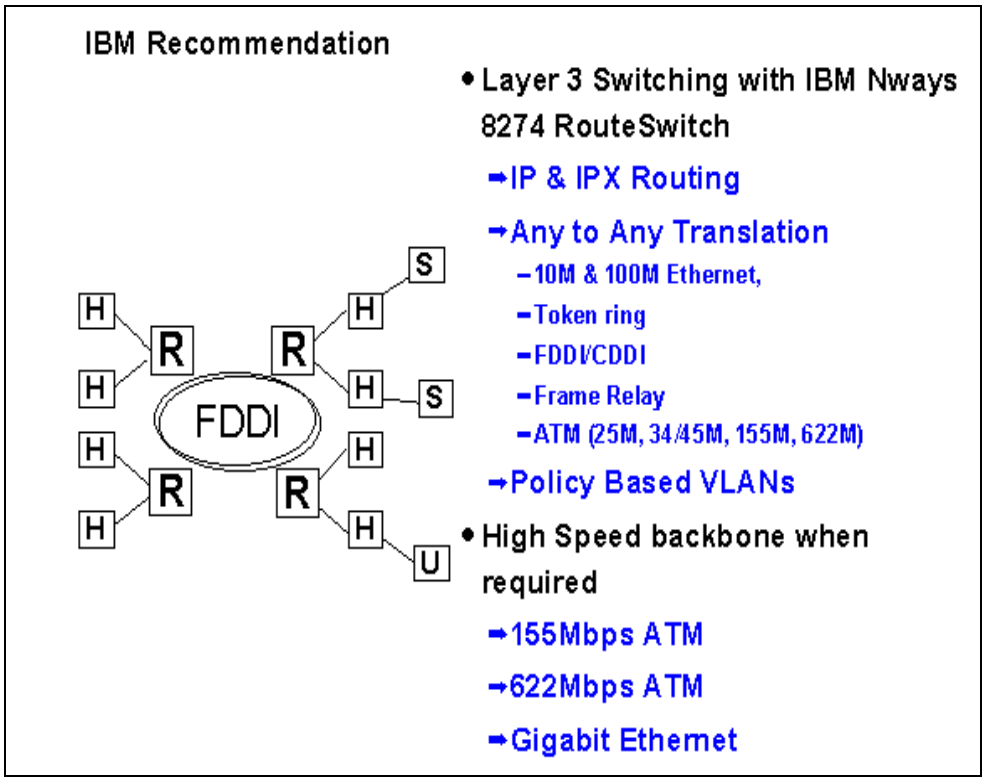


Figure 66. Better performing redesigned network

The layer 3 switching solution solves the immediate problem of internetwork segment congestion. It also provides more flexibility. We can create virtual LANs (a collection of devices grouped within their own LAN broadcast domain) to minimize LAN broadcasts and create logical networks across physically separate segments. Switching occurs within the virtual LAN at the MAC level (OSI layer 2). Routing occurs outside the virtual LAN at the network level (OSI layer 3).

6.7 Capacity and response time sizing: Security

Chapter 5, “Security implications and performance” on page 79, discusses security features that you may need to deploy in your environment. As you may suspect, security comes at a certain price to performance. Some of this is a one-time event, such as a user having to enter a user ID and password or selecting a client certificate at your browser. These add delay time, but are quite visible so the end user often accepts it as a small price to pay.

We must plan appropriately for recurring or multiple events, such as firewall proxies or encryption of user transactions between the browser and the server. This has an affect on the server, network, and sometimes the client.

6.7.1 SSL environment

Chapter 5 gives an in-depth description of key technologies in enabling secure transactions over a private or public network, including digital certificates and Secure Sockets Layer. As you may expect, there is additional processing overhead on the client and server but, fortunately, the bulk of it is during the initial handshaking. Nevertheless, you must plan appropriately for its usage in our client and server load. Some additional network traffic is incurred with secure data exchange.

From a server perspective, this is well documented in the appropriate AS/400 Performance Capabilities Reference, such as our examples in 6.2, “Sizing for different application characteristics” on page 91. To illustrate this, the relative uplift factor for a 40-bit RC-4 SSL over a non-secured page is shown in Table 17.

Table 17. Relative uplift for a 40-bit SSL encryption: AS/400 HTTP server

OS/400 Release	V4R2	V4R3	V4R4
Static page without cache uplift	2.3	2.3	2.5
Net.Data page with SQL uplift	1.3	1.1	1.1

Note that the server uplift factor is most pronounced on static pages. What does this mean to overall response time? The answer depends on the type of pages being served, the encryption level, the client, and the network load. We used the technique outlined in 6.5.1.1, “Web page response time” on page 107, to measure the overall response time at the browser. This allows us to compare the response time at the client between a secure and non-secure page. The same pages referenced in 6.5.1.2, “Measuring the client contribution” on page 109, were used. At the client, we used the desktop and laptop PC workstations cited earlier and a browser with 128-bit SSL encryption. The test was run in a LAN environment with a 16 Mbps Token-Ring. A quick test showed us the results in Table 18 for static Web pages.

Table 18. 128-bit SSL impact on overall response time

Scenario	86 KB page	109 KB page	139 KB page	569 KB page
Local laptop PC	1.15 seconds	2.3 seconds	2.75 seconds	10.22 seconds
Network laptop PC	1.38 seconds	2.47 seconds	2.91 seconds	18.59 seconds
Network and 128-bit encryption laptop PC	1.65 seconds	3.02 seconds	3.35 seconds	26.42 seconds
Incremental uplift	1.20	1.22	1.15	1.42
Local desktop PC	.17 seconds	.34 seconds	.39 seconds	3.52 seconds
Network desktop PC	.38 seconds	.59 seconds	.69 seconds	3.98 seconds
Network and 128-bit encryption desktop PC	.42 seconds	.81 seconds	.82 seconds	5.2 seconds
Incremental uplift	1.09	1.37	1.20	1.31

If we look at the laptop PC analysis, we calculated earlier that the network and server (primarily network) impact averaged out to a 29% uplift factor. If we then factor in the 128-bit SSL, this adds an additional 25% uplift, if we calculate a

simple arithmetic average. Combined, this shows that a 128-bit SSL transaction over our LAN adds over 60% to the client-only load time.

If we look at the desktop PC analysis, we calculated earlier that the network and server (primarily network) impact averaged out to a 70% uplift factor. If we then factor in the 128-bit SSL, this adds an additional 24% uplift, if we calculate a simple arithmetic average. Combined, this shows that a 128-bit SSL transaction over our LAN adds about 110% to the client-only load time.

In a wide area environment with lower bandwidth and higher latency, this percentage would be lower since the network contributes an even higher share to the total response time. As we have stated many times, the results in your environment may vary substantially. Certainly, whether the environment is a LAN or WAN, SSL and encryption can add substantially to the response time and infrastructure load.

6.7.2 Firewall and proxy environment

Chapter 5, “Security implications and performance” on page 79, discusses the concept of firewalls. Firewalls provide essential security services: IP packet filtering, application level proxying, and SOCKS proxying capabilities. Our focus is on the latter two services. Most users in a corporate environment access the Internet through a proxy or SOCKS server. This provides a barrier between a secure and unsecure network.

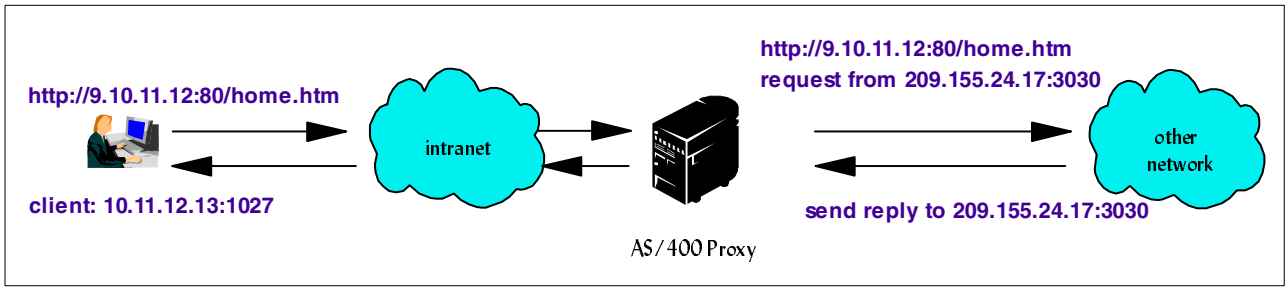


Figure 67. Example of an HTTP proxy server

In the example in Figure 67, the browser client actually maintains its connection with the AS/400 proxy. The AS/400 proxy maintains two sets of connections: proxy to browser and proxy to requested host. This capability has been available on the AS/400 HTTP server since V4R3. In addition to providing a more secure session for our browser users, proxy servers can cache Web content. For example, you may want to have frequently accessed Internet pages cached locally, which gives your users a better response time because the Web page and graphics files would be retrieved from a local server. This would offer a tremendous opportunity to reduce WAN bandwidth requirements. This has also been available on the AS/400 server since V4R3.

The trade off, of course, is that your server must incur some amount of processing to act as a proxy. It must act as an application router, managing the connections between itself, the client, and the server. It also may have to incur an amount of overhead to resolve the client’s domain name server request.

If you use your server as a proxy cache, you also must plan for a certain amount of processing necessary to serve Web pages from its own file system (the AS/400 QOPENSYS file system).

6.7.2.1 Example: Proxy server only

This section goes through a process to determine the server load impact if you choose to use your AS/400 system as a proxy server (Figure 68). In this example, an AS/400 system is acting as an application router.

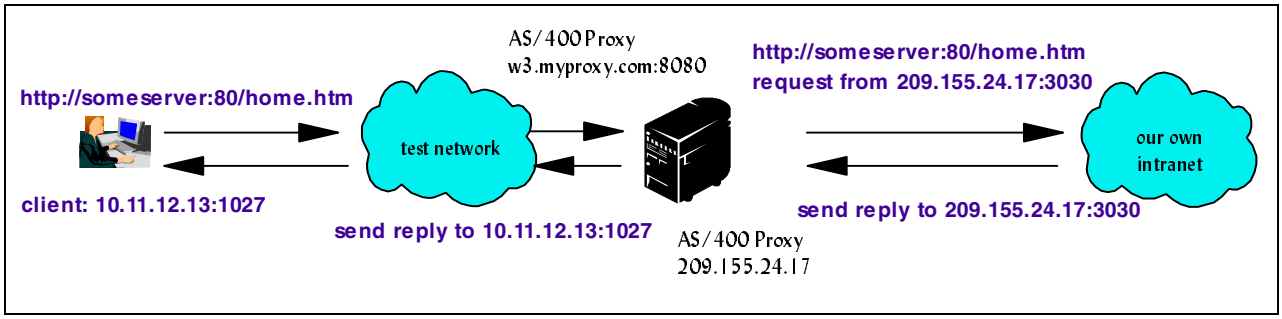


Figure 68. AS/400 acting as a proxy server

The client on the test network configured the browser to use the HTTP proxy w3.myproxy.com on port 8080. The client can then access Web resources on your intranet, using the AS/400 Web proxy server. The client does not have access to the intranet domain name server (DNS), so the AS/400 proxy must do the DNS lookup on behalf of the client. The AS/400 proxy server does *not* need to have the DNS server active to provide this function. Similarly, the AS/400 proxy server does *not* need to have the IP datagram forwarding option turned on (Figure 69).

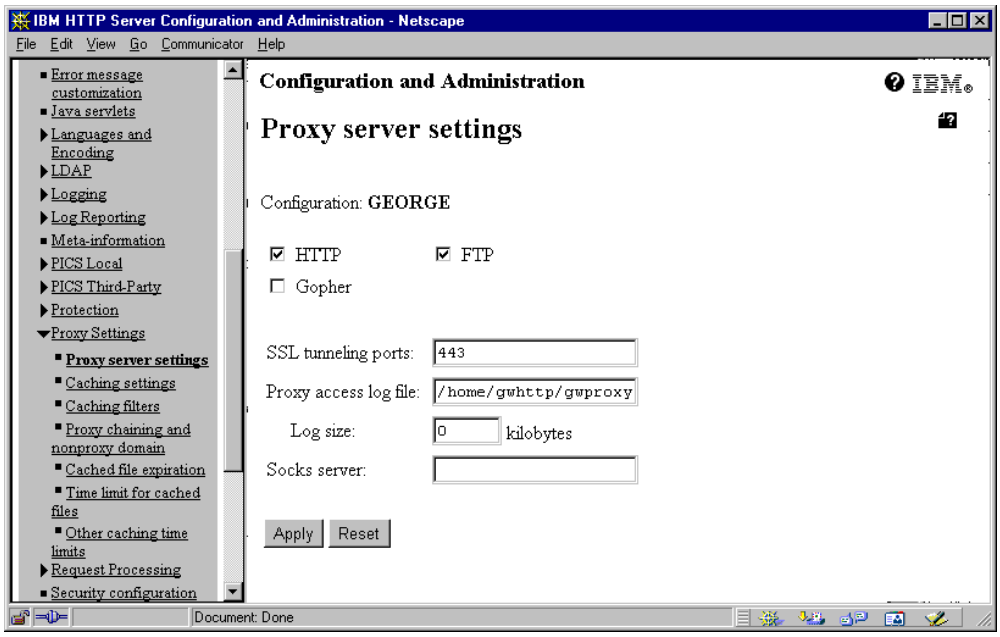


Figure 69. Configuring the AS/400 as a proxy server

In examples such as this, a commercially available Web load generation tool, such as Platinum Final Exam WebLoad from Platinum Software Corp., or

SilkPerformer from Segue Software Inc., is helpful for sizing. For this test, we used the RSW eTest Suite from RSW Software Inc. to generate a multiple client simulation (Figure 70). The test involved repetitively accessing 10 different Web pages scattered across our intranet (some LAN and some WAN connected). The 10 pages had a total of 100 objects, fairly evenly distributed.

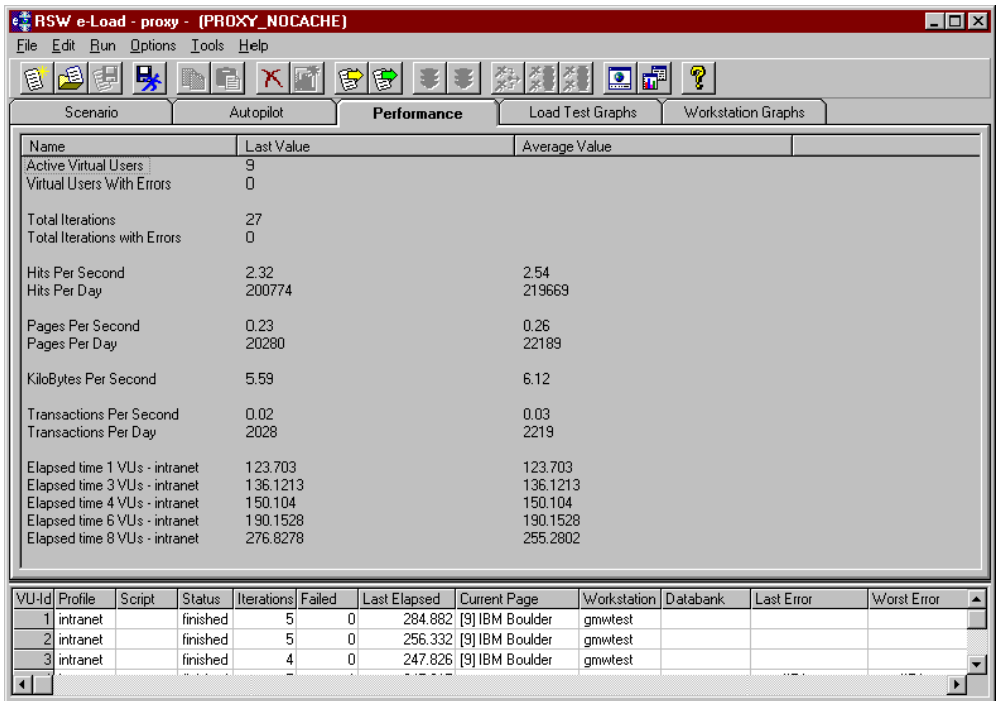


Figure 70. Simulating multiple clients using the AS/400 system as an HTTP proxy

We enabled proxy logging to allow us to monitor the number of hits and amount of data transferred. We also used the AS/400 Performance Monitor to record the server workload impact (`STRPFRMON`).

After we gathered a sufficient amount of data, we ended the AS/400 Performance Monitor (ENDPFRMON) command. Next, we used the AS/400 `GO PERFORM` command to get the AS/400 Performance Tools screen. Then, we selected option **7** to view performance data for our particular HTTP server job. The screen shown in Figure 71 appeared.

```
                              Display Jobs

Elapsed time  . . :   00:26:23          Member  . . . . . :   Q992431016
                                         Library . . . . :     QPFRDATA

Type options, press Enter.
  5=Display job detail

                                          Job      CPU      Tns      Avg     Disk
Option  Job         User        Number    Type     Util     Count    Rsp     I/O
        GEORGE      QTMHHTTP    054618    BCH      .38          0     .0      69
```

Figure 71. Excerpt from Performance Monitor job display

During this 26 minute monitoring interval, the AS/400 Web server that was proxy serving had an average CPU utilization of .38%. Our next steps were to correlate this CPU utilization to the requests.

The eTest Suite software table showed an average of about 2.5 hits per second. We used the proxy log file on the AS/400 system for a more in-depth analysis. We used the NetIntellect product to analyze our proxy server log for the number of hits and data transfer (Table 19). Our only traffic on the proxy server during the 10:00 a.m. interval was during the AS/400 Performance Monitor data collection.

Table 19. Proxy log results analysis

Summary of Activity by Time Increment				
Time Interval	**Hits**	**Page Views**	**KBytes Transferred**	**User Sessions**
08:00:00 - 08:59:59	175	22	855 K	1
09:00:00 - 09:59:59	861	431	3,755 K	0
10:00:00 - 10:59:59	3,840	528	23,446 K	0
11:00:00 - 11:59:59	5,629	644	34,163 K	2
12:00:00 - 12:59:59	6,268	683	36,730 K	0
13:00:00 - 13:59:59	0	0	0 K	0
14:00:00 - 14:59:59	1	0	30 K	1
Total	**16,774**	**2,308**	**98,979 K**	**4**

The most important statistics, for our purpose, were the 3,840 requests (object hits), 528 page hits, and 23.4 MB of data over the 26 minutes in which we ran the test. After doing the math, we received 2.5 object hits per second, .15 page hits per second, and a data throughput of about 120 Kbps. This was with .38% utilization of our server (AS/400 Model 170, with Feature 2388 and a CPW rating of 1090). We can use this as a rough approximation for sizing the impact of using our AS/400 system as a proxy server for this particular set of pages. Certainly, this will vary for your particular environment. See the following formula:

```
2.5 hits/second / (1090 CPW * .38%) = .6 hits/second/CPW
```

6.7.2.2 Proxy caching server example

The next step is to assess the impact of a proxy caching server. To carry the prior example a step further, we want to cache these Web pages on our local server. Many of these pages are on servers that are not part of the local network. They are at other sites connected with lease lines or Frame Relay wide area connections. To improve end user response time, we want to cache these pages on the local AS/400 system. We set the AS/400 server as a proxy cache, using all default values (Figure 72).

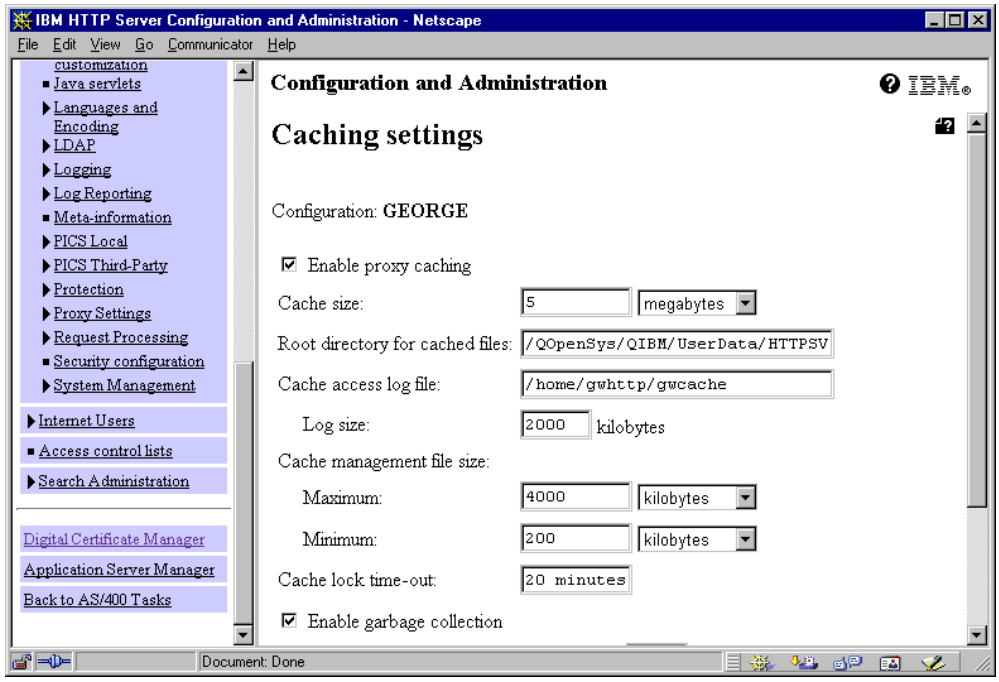


Figure 72. Enabling the AS/400 proxy server for caching

Our client simulation test is identical to the test done with just the proxy server. We started AS/400 Performance Monitor to record the server workload impact (STRPFRMON).

We replicated the previous test for the proxy cache. This is to enable proxy caching on the AS/400 system.

After we gathered a sufficient amount of data, we ended the AS/400 Performance Monitor (ENDPFRMON). Next, we used the AS/400 GO PERFORM command to access the AS/400 Performance Tools screen. We selected option **7** to view the performance data for our particular HTTP server job (Figure 73).

```
                                Display Jobs

Elapsed time   . . :   00:30:04          Member  . . . . . :   Q992431119
                                           Library . . . . :     QPFRDATA

Type options, press Enter.
  5=Display job detail

                                                 Job      CPU      Tns      Avg     Disk
Option  Job          User         Number  Type    Util     Count    Rsp      I/O
        QEJBADMIN    QEJB         054707  BCH     1.56         0     .0     1412
        GEORGE       QTMHHTTP     054687  BCH     1.10         0     .0    91373
```

Figure 73. Excerpt from the Performance Monitor job display: 11 a.m. test

We ran this test twice. The results from time segment 2 are shown in Figure 74 on page 122.

```
                                Display Jobs

Elapsed time  . . :   00:31:47          Member  . . . . . :   Q992431159
                                         Library . . . . :     QPFRDATA

Type options, press Enter.
  5=Display job detail

                                          Job      CPU      Tns     Avg    Disk
Option  Job         User        Number    Type     Util     Count    Rsp     I/O
        GEORGE      QTMHHTTP    054687    BCH      1.52         0     .0  132457
```

Figure 74. Excerpt from the Performance Monitor job display: 12 p.m. test

We combine these two segments to calculate the hits per second, data throughput, and estimated hits/second/CPW. looking at Table 19 on page 120, we see that for the 11 a.m. segment, there were 5,629 requests (object hits), 644 page hits, and 34.2 MB of data over the 30 minutes in which the test was run. We also see that for the 12 p.m. segment, there were 6,268 object hits, 683 page hits, and 36.7 MB of data over the 31 minutes in which the test was run.

Doing the math for segment 1, we get 3.1 object hits per second, .36 page hits per second, and a data throughput of about 152 Kbps. This was with 1.1% utilization of our server (AS/400 Model 170, Feature 2388 and a CPW rating of 1090).

If we repeat this for segment 2, we get 3.3 object hits per second, .36 page hits per second, and a data throughput of about 153 Kbps. This was with a 1.5% CPU utilization of our server. Using an average for these two scenarios, we can again calculate a relative estimate as follows:

```
3.2 hits/second / (1090 CPW * 1.3%) = .23 hits/second/CPW
```

We can use this as a rough approximation for sizing the impact of using our AS/400 system as a caching proxy server for this particular set of pages. Certainly, this will vary for your particular environment. Note that there is a significant difference in server load (a relative factor of .6/.23) when we add the proxy serving capability to our server.

The objective with proxy caching is that we improve the response time to our end users. We performed a quick test to access these test pages—without any proxy server and with a proxy caching server. Our results are shown in Table 20.

Table 20. Response time improvements with proxy cache

Load time without proxy (seconds)	Load time with proxy cache (seconds)
4.5	4.5
2.6	1.9
4.8	4.8
6.4	4.0
7.0	5.0
17.8	13.7
3.3	3.7

Load time without proxy (seconds)	Load time with proxy cache (seconds)
6.0	4.4
6.6	4.7
17.1	12.5

Note that, in most cases, we achieved a moderate decrease in overall response time at the client. Again, variations in the network traffic and server load can have a sizable impact. If we add up the load times without the proxy and with the proxy cache, we get 76.1 seconds and 59.2 seconds respectively. Therefore, we reduce the response time by about 22%. We need to point out that these are all static Web pages and certainly this will not help on dynamically generated pages.

6.8 Sizing tools

Since the concept of sizing our server, client, and network involves so many unknowns, it is difficult to expect a standard, easy-to-use chart or tool to do an appropriately detailed analysis. This section highlights a couple of options you may want to consider.

6.8.1 AS/400 Workload Estimator

IBM has a browser-based tool available for Business Partners to use as a quick approximation for AS/400 server workload estimation. Authorized IBM Business Partners can access this from the Internet at: `http://partners.boulder.ibm.com`

This site requires registration and the actual estimator application requires that a user ID and password be entered. This tool can be used to predict the server load for four workload types:

- Java applications
- Lotus Domino applications
- IBM Net.Commerce applications
- Traditional applications

The following steps show you an example of using this tool to estimate AS/400 requirements for traditional applications:

1. Specify the workload type (Figure 75).

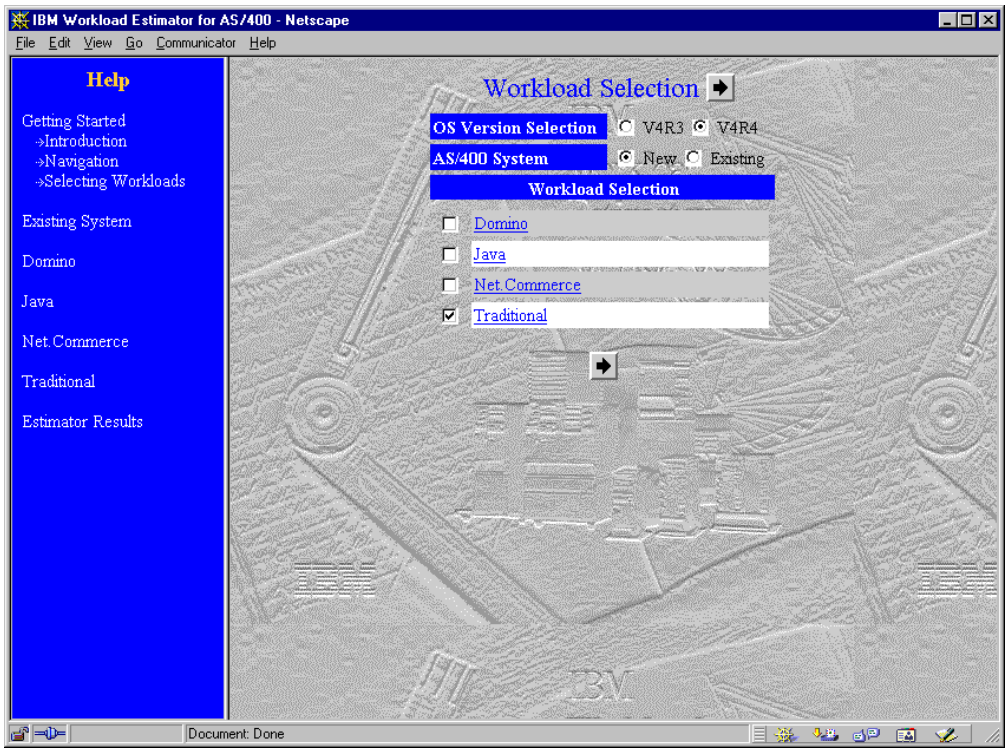


Figure 75. Workload Estimator selection criteria

2. Specify the expected workload quantity (Figure 76).

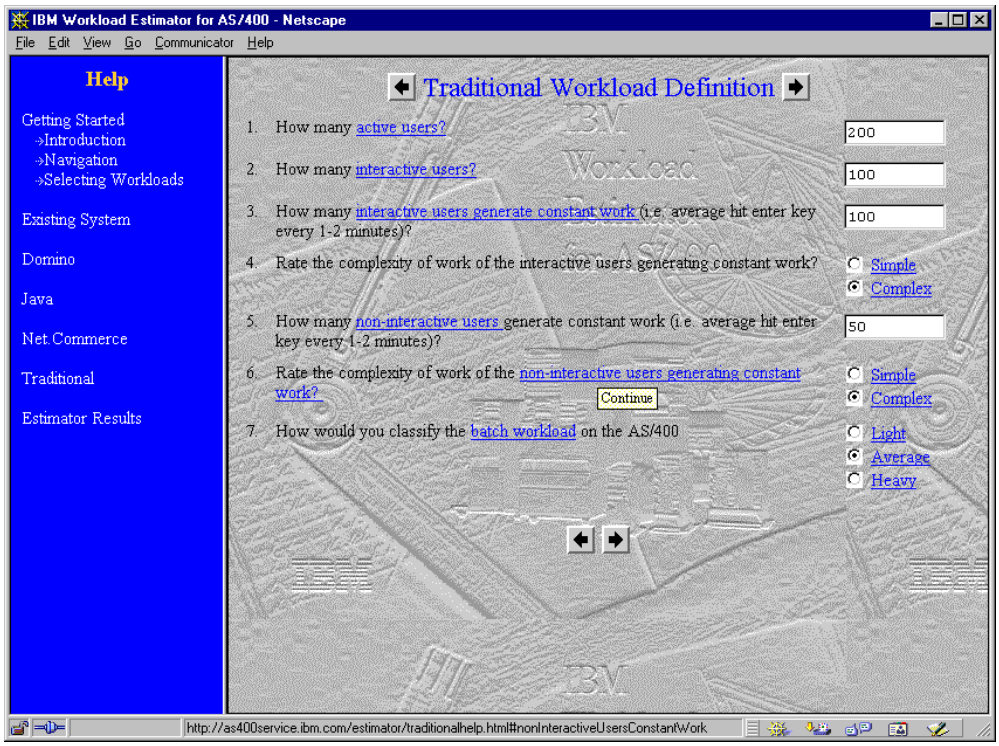


Figure 76. Workload Estimator sizing criteria

3. Review the results as shown in Figure 77.

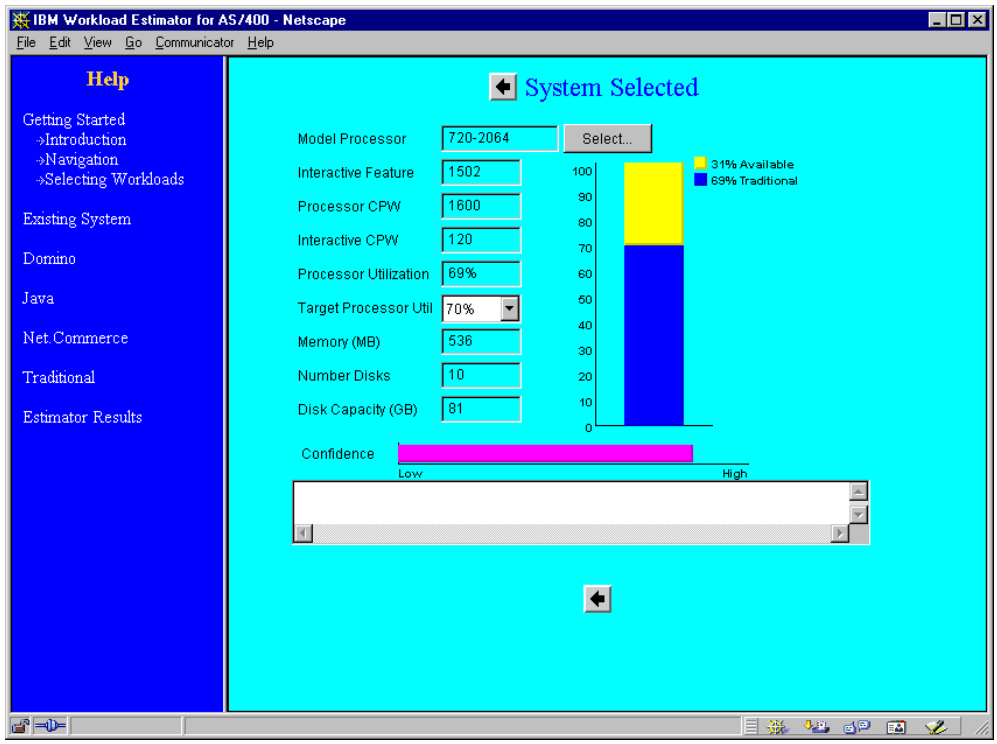


Figure 77. Workload Estimator system recommendation

6.8.2 BEST/1 Sizer

The BEST/1 component of the AS/400 Performance Tools Licensed Program Product provides a full function, robust sizing and analysis tool for many types of AS/400 workloads. Such workload types include interactive, batch, and client/server, as well as for HTTP serving. It enables the user to define system resources (AS/400 model, CPU, main memory, disk and communications IOPs) and system workloads. It also allows the user to analyze the potential results. A simple example of using the BEST/1 product to size an HTTP server load on an AS/400 Model 170 is shown in the following steps:

1. Start the BEST/1 tool `STRBEST` command.
2. Enter option `1` (Work with models).
3. Enter option `1` (Create), and give your model a name.
4. Enter option `2` to create a user-defined workload.
5. Enter option `10` to configure an AS/400 server for disk arms, communications IOAs and lines, ASPs, and storage pools. Figure 78 on page 126 shows an example.

```
                                Configuration

 CPU Model  . . . . . :   2385      Comm IOPs  . . . . . :   0
 Main stor (MB) . . . . :   3584      LAN lines  . . . . . :   2
 Main stor pools  . . . . :   4       WAN lines  . . . . . :   0
 Disk IOPs  . . . . . :   0
 Disk ctls  . . . . . :   0         Multifunction IOPs . . . . :   1
 Disk arms  . . . . . :   9         Disk IOAs  . . . . . :   2
 ASPs . . . . . :   1         Comm IOAs  . . . . . :   2
                                    IPCS IOAs  . . . . . :   0

 Select one of the following:

      1. Change CPU and other resource values
      2. Work with disk resources
      3. Edit ASPs
      4. Edit main storage pools
      5. Work with communications resources

 Selection or command
 ===>
 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Check configuration
 F17=Correct configuration          F24=More keys
```

Figure 78. Selecting an AS/400 configuration in BEST/1

6. After creating your model system, press Enter to return to the Work with BEST/1 Model screen. Enter option `1` (Work with workloads).
7. Press the F9 key to retrieve predefined workloads.
8. Enter `1` in the SERVER workload group field to add IBM supplied client/server workloads.
9. Enter `1` in the WWW STATIC PAGE SERVER field to add an HTTP server workload.
10. Enter a name for the HTTP workload and the transactions/hour/user.

```
                            Create Workload

Type changes, press Enter.
  Workload  . . . . . .   WEB_PAGES    Name
  Workload text . . . . .   WWW STATIC PAGE SERVER
  Workload type . . . . .   *NORMAL      F4 for list
  Usage mode  . . . . . .   4            1=Casual, 2=Interrupted, 3=Steady,
                                           4=N/A


                         Functions  Avg K/T  ------Tns per Function------
Function Text             per User   (secs)       Inter        Non-inter
WWW STATIC PAGING            50.00      N/A           .00          59.99
```

Figure 79. Creating a workload in BEST/1

11. After creating your workloads, press Enter to return to the Work with BEST/1 Model screen. Enter option `2` (specify objectives and active jobs).
12. Enter the number of active jobs for each specified workload (Figure 80).

```
                    Specify Objectives and Active Jobs

 Model/Text:   HTTPSIZING  george's test http sizing

 Type changes, press Enter.

                     Workload    Active  ----Interactive-----   Non-inter
 Workload    Connect Type         Jobs   Rsp Time     Thruput    Thruput
 DTQ         *LAN    *NORMAL      10.0       .0           0          0
 WEB_PAGES   *LAN    *NORMAL     100.0       .0           0          0
```

Figure 80. Specify Objectives in BEST/1

13. After entering the job activity level, press Enter to get back to the work with BEST/1 model screen. Enter option `5` (Analyze the current model).

14. Analyze the results (Figure 81), and repeat as necessary for sizing different workloads, machine types, disk arms, communications IOPs, and other "what-if" scenarios.

```
                        Display Analysis Summary

CPU Model / release level  . . . . . :   2385        V4R4M0
Main Storage . . . . . . . . . . . . :   3584       MB

                                        Quantity    Predicted Util
Multifunction IOPs . . . . . . . . . :     1            71.3
Disk IOAs  . . . . . . . . . . . . . :     2            37.7
LAN IOAs . . . . . . . . . . . . . . :     2              .0
WAN IOAs . . . . . . . . . . . . . . :     0              .0
Integrated PC Server IOAs  . . . . . :     0              .0
                                                                  More...
                                           Interactive   Non-interactive
CPU utilization % . . . . . . . . . . . :   .0              17.6
Transactions per Hour . . . . . . . . . :   0               165087
Local response time (seconds) . . . . . :   .0              .0
LAN response time (seconds) . . . . . . :   .0              4.9
WAN response time (seconds) . . . . . . :   .0              .0
```

Figure 81. Displaying analysis summary in BEST/1

6.9 Considerations for estimating peak loads

If it is difficult to predict an average workload for a Web or intranet site, what about the peak workload? Most organizations do not have the luxury of having the money and other resources needed to plan for millions of transactions per day, regardless of the actual traffic. If you log access to your server, you can analyze the statistics to estimate a peak load by asking these questions:

- Are my hits strictly from 6 a.m. to 6 p.m. Pacific time, or fairly well spread across all 24 hours of the day?
- Do I have a higher number of hits on Tuesdays (the day our Web site announces weekend air fare reductions) or on the last day of the month?
- Does my traffic depend on other predictable events, such as product announcements or earnings reports?

Since our subject is sizing, we assume you have no historical data. You may need to estimate a particular expected service level and make periodic adjustments as necessary. You should also try to assess if traffic is totally random or based on specific events or factors mentioned previously. Generally, you should rely on intuition and common sense over complex mathematical formulas.

If you are seeking a mathematical approach, the Poisson statistical distribution can be used to approximate the probability of *x* events happening within a particular time interval, given an average number of events *y*. For example, if your site gets an average of 10 hits per second and you plan for 15 hits per second, is there a high likelihood that your site will exceed 15 hits?

If you choose this approach, consult text or references on probability and statistics. Table 21 shows an example of using 10 and 20 hits per second, and a relative "safety factor".

Table 21. Estimating the peak hits/second with a Poisson distribution

Expected hits/second	80% safety factor	90% safety factor	95% safety factor	99% safety factor
10	12	14	15	17
20	23	25	27	30

To cite our earlier example of the impact of planning for 15 hits per second when our average is 10 hits per second, we predict 95% of the time the hits per second will not exceed 15. In this case, we uplift our average by 50% (15/10). For our 20 hits per second scenario, for a 95% confidence feeling, our uplift is 35% (27/20). Note that this is not a linear factor. As our average number of hits per second increases, the uplift factors decrease.

Certainly, mathematical approaches such as this must be applied with a certain amount of caution.

6.10 Summary

This chapter covered many diverse and related topics. The intent is to provide you with a methodology in sizing resources contributing to overall Web page response time, such as server, client, and the network. Our analyses were focused on individual components, such as the AS/400 server and the workstation. Despite the moderate amount of work, you may find these methods the most accurate and flexible means of determining the impact of your Web applications and traffic, and for specifying an appropriate solution to your customers or end users.

We discussed the complexity of several key facets in providing browser-based information solutions. We covered application characteristics, server resources, such as CPU, disk arms, memory, communications IOPs, workstation capabilities, the networking infrastructure, and security considerations. Each contributes to the most important, and most difficult to analyze metric of all, end-to-end response time for the user.

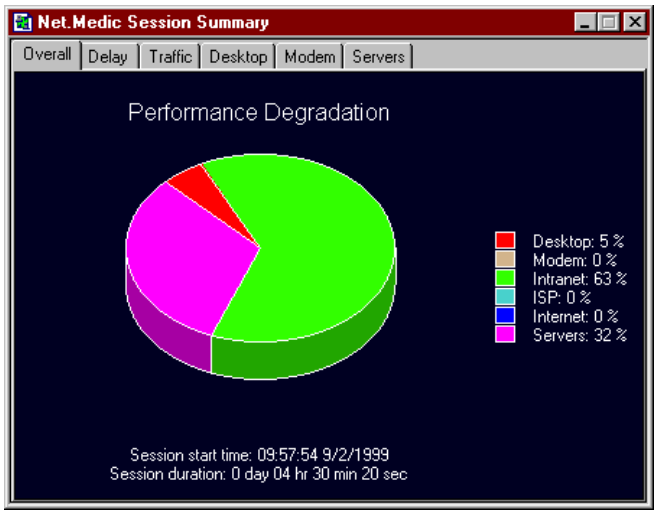


Figure 82. Example breakdown of response time components from a PC-based tool

Figure 82 shows an example of a workstation-based tool that uses historical information and inferences to determine Web page response time contributors for the areas we discussed, including client workstation, server, network, and any gateways. These tend to be easy to setup and use and can be helpful in assessing your applications and network infrastructure.

Workstation-based products such as this give a good summary of what is happening at the client, including overall response times and the network bandwidth realized during page downloads. They do not give a detailed view of what is happening at the server, since they cannot analyze server load or considerations reflecting dynamic pages and real business applications. Similarly, server-based tools provide an enormous amount of valuable information, such as resource utilization and application level results. However, they do little in terms of the client response time or network considerations. You will find that the best solutions involve a combination of client-based tools, server-based tools, and network-specific tools.

Sizing will be an ongoing task in your Web and e-business application enabling projects. New applications, workloads, and scenarios will be common. New equipment and the money to deploy these solutions will be less common. Therefore, it is to your advantage to do as good of a job as possible in sizing these new opportunities.

Chapter 7. Capacity planning for Web-based applications

Earlier chapters show you the wealth of information that can be obtained from HTTP server access logs. Such information includes the most frequently accessed resources, activity by day and time interval, amount of data sent and received, IP addresses of those accessing our site, and the number of actual hits. In this chapter, we see how the access log and other logs can be used to your advantage for ensuring that your Web application environment can be scaled to accommodate additional traffic and workloads in an affordable manner. Another critical component is the network and communications infrastructure and ensuring the resources under your control are not a performance impediment. We also show how the AS/400 Performance Tools and BEST/1 product can be used to model many "what if" scenarios. Plus, we recommend server resources that are necessary to meet our requirements.

7.1 Capacity planning basics

Chapter 6, “Sizing Web-based applications” on page 91, deals with sizing the client, network, and server resources. The premise behind sizing is that you contemplate a brand new workload, or a substantial increase in an existing workload, and make assessments and calculations to determine your resource requirements. Expressed another way, there is zero or little real data to use. On the other hand, capacity planning comes into play on a regular basis as you fine tune your Web application infrastructure and have to plan more extensively for peak requirements, additional workloads, and growth. In this section, we discuss the ongoing process of measuring resource utilizations and planning for the future. Think of capacity planning as developing estimates of server, network, and client resources required to deliver a quantifiable performance level in accordance with a formally or informally documented service level agreement. It is based on a forecasted level of business activity at specific future dates.

For an excellent resource on general AS/400 capacity planning concepts and a practical introduction to AS/400 Performance Monitor and BEST/1, refer to *AS/400 Server Capacity Planning*, SG24-2159.

7.2 Capacity planning for different application characteristics

In Chapter 6, “Sizing Web-based applications” on page 91, we show several examples of sizing for different application characteristics, such as static pages and dynamic pages with Net.Data. This section shows how to analyze Web page traffic and determine how close the actual results compare with what we planned for during the sizing phase. For example, if your site is getting a higher percentage of dynamic page requests, such as account inquiries, you will have a greater server load than anticipated.

7.2.1 Categorizing the page hit type

The AS/400 HTTP server access logs are essential in helping to understand the types of requests hitting the server, and the types of pages and documents being requested. For example, these two log entries represent a request for a static page containing an HTML form (GET method), and a form submission representing a dynamically generated Web page (POST method):

```
9.5.62.190 - - [27/Aug/1999:15:56:44 +0500] "GET /democgi HTTP/1.0" 200 1182
9.5.62.190 - - [27/Aug/1999:15:57:10 +0500] "POST /cgibin/cgiprog HTTP/1.0" 200
1034
```

The log analysis tools give us a breakdown of dynamic pages and forms, since they count hits as being the POST method or a GET method with a question mark (?) in the URL string. Figure 83 uses a graphical representation to show dynamic page hits tracking.

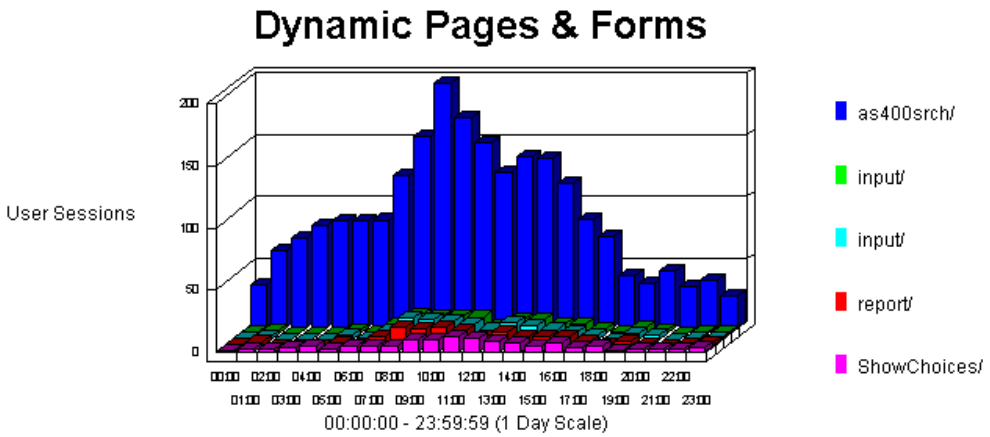


Figure 83. Graphical example of dynamic page hits tracking

Table 22 presents another way of viewing dynamic page hits tracking.

Table 22. Example of dynamic page hits tracking

Dynamic Pages & Forms				
	Dynamic Pages	No. of Pages	% of Total	User Sessions
1	/cgi-bin/as400srch/	4,139	66.7%	2,183
2	/net.data/rmedia/rmosverf.d2w/input/	484	7.8%	125
3	/net.data/rmedia/rmosinfo.d2w/input/	407	6.55%	136
4	/net.data/rmedia/rmoshome.d2w/report/	305	4.91%	127
5	/howtobuy/ShowChoices/	141	2.27%	127
6	/QSYS.LIB/SNIPPETS.LIB/sendfile.pgm	120	1.93%	46
7	/e-solution/Processesolution/	90	1.45%	56
8	/servlet/feedspd	78	1.25%	42
9	/net.data/tstudio/workshop/snippets/newsnip.mac/search/	73	1.17%	43
10	/net.data/rmedia/rmoshome.d2w/input/	69	1.11%	44

The example in Table 22 shows that there were about 6,200 (4139/.667) "dynamic page hits". If the total page hits (page views, or impressions) is 62,000, you can easily see that dynamic pages represent 10% of the total traffic. Granted, this is a one-day snapshot, the process is essentially the same for a longer interval.

7.2.2 Categorizing the page objects content

You should also investigate the number of objects per page and the total size of the page hit to determine how closely this matches our estimates. Again, you can use the access log to help with the analysis.

The WebTrends tool offers two sets of download statistics. The first set is the total for the site, regardless of client caching. The second set is the total factoring in client caching. This is important to note, since the logging reports indicated that about 21% of the hits were serviced from cache. To further illustrate this concept, the total site traffic statistics showed about 339,000 object hits, 125,000 page

hits, and about 3,695 MB of data. This would imply that, on average, the site pages in our example are a document, with 1.7 other elements ([339 KB - 125 KB/125 KB), and the average page is 30 KB (3,695 MB/125,000 hits). Let's see whether this is valid.

Table 23. Tabular example of served objects

Most Downloaded File Types			
	File type	Files	K Bytes Transferred
1	gif	127,990	531,370
2	htm	107,155	1,365,939
3	css	10,481	15,105
4	html	7,668	119,940
5	jpg	5,927	81,334
6	js	2,678	26,994
7	d2w/input	1,229	15,804
8	rpm	808	60
9	ra	554	57,572
10	pdf	449	783,250
	Total Files & K Bytes Transferred	**264,939**	**2,997,364**

If you look at the numbers in Table 23, you see that the number of files and kilobytes transferred is about 80% of the total. It does not factor in cached files. If you look at the base pages as being file type htm, html, and the d2w/input, you can add this up and get 116 KB base pages. If you exclude the PDF file types (these are separately downloaded), you get about 148 KB other objects. This results in the average document being a base page, with 1.3 other elements per page (148 KB/116 KB), and the average page is 19 KB. Again, we emphasize that this does not include any cached pages. Also, note that a relatively small number of hits for PDF documents accounted for over 25% of the download traffic.

What can we conclude from these analyses? They indicate:

- About 90% of the hits are for static pages, and 10% are for dynamic pages.
- The "average Web page" download is of the parent document, with about 1.5 additional files and about 25 KB (if you factor out the PDF data from the total).
- Client caching has offered a substantial benefit by serving over 20% of the total requests.

You can use this data to recalculate the static and dynamic Web page hit data from the sizing exercise, if necessary. You can also use this information, and intuition, to estimate future increases in dynamic Web page hits as a percentage of total hits, as more business transaction capabilities are added to your Web site.

7.2.3 Correlating server CPU utilization

The previous two sections give a good view of how customers are accessing the AS/400 HTTP server. During the sizing exercise, we estimated the traffic based on the number of hits, the size of the page and number of graphics, and the breakdown in static and dynamically generated pages for a particular scenario. We can compare the plan to the actual results and make modifications, if necessary. We must also compare the AS/400 server load's plan to the actual results and determine if modifications are necessary.

We use the AS/400 Performance Monitor data in conjunction with the logging data to determine how the HTTP server responds to the actual traffic observed. In the previous two sections, we observed about 339,000 object hits, 125,000 page

hits, and about 3,695 MB of data total workload at the Web site. Client caching contributed to a significant reduction in actual file downloads. Let's look at the AS/400 server utilization during this period.

From the AS/400 Performance Tools main menu, enter the (GO PERFORM) command and complete this process:

1. Enter option 7 (Display performance data).
2. Move to the appropriate performance data set, and entert option 1.
3. Press F13 to select all intervals measured. Then press the Enter key.
4. After the report is generated, you can observe the results. Press F6 to display all jobs.
5. Look for your job name (the name of your HTTP server instance) and user QTMHHTTP.

You should see a results screen similar to the one shown in Figure 84.

```
                                Display Jobs

Elapsed time  . . :   23:42:14          Member  . . . . . :   Q992280000
                                          Library . . . . :     HTTPPERF

Type options, press Enter.
  5=Display job detail

                                          Job      CPU      Tns      Avg     Disk
Option  Job         User        Number    Type     Util     Count    Rsp      I/O
        HTHJOBDS    MS773       485225    BCH       .00         0     .0      647
        HTHJSAUT    MS773       485226    BCH       .00         0     .0     8204
        HTHUSRACC   MS773       485227    BCH       .00         0     .0     2519
        HTHUSRDEL   MS773       485228    BCH       .00         0     .0     2329
        HTTP80      QTMHHTTP    483159    BCH      3.34         0     .0   ++++++
        HTTP80      QTMHHTTP    483163    BCH       .36         0     .0   116320
        HTTP80      QTMHHTTP    483172    BCH       .00         0     .0      429
        HTTP80      QTMHHTTP    483425    BCH       .15         0     .0    52361
        HTTP80      QTMHHTTP    483792    BCH       .08         0     .0    27264
        HTTP80      QTMHHTTP    483793    BCH       .03         0     .0    12502
```

Figure 84. HTTP server CPU usage

If you add the CPU utilization for all of the Web server jobs, with HTTP80, you get a total during the interval of 5.3% (there are numerous job entries not shown in Figure 84).

In summary, we determined that the site had 339,000 object hits, 125,000 page hits, and about 3,695 MB of data total workload at our Web site. Client caching reduced the actual downloads by about 20%. The AS/400 HTTP server used a bit over 5% to service this workload. You can use this data to fine tune sizing estimates as you prepare for increased traffic and new applications at your Web site.

7.3 Capacity planning for AS/400 resources

The example in 7.2, "Capacity planning for different application characteristics" on page 131, introduces the concept of capacity planning and assessing AS/400

server CPU utilization. However, our analysis was restricted solely to the HTTP serving workload. In reality, the AS/400 server may be running a variety of additional workloads, even 5250 interactive applications. You must also factor in other AS/400 resources such as disk arms, main memory, and communications IOPs.

7.3.1 Main processor

Chapter 6, “Sizing Web-based applications” on page 91, briefly discusses AS/400 system models, AS/400 server models, and the AS/400e 7xx server models. AS/400 system models are meant for balanced workload requirements, including batch jobs, 5250 interactive applications, and a modest amount of client/server jobs, such as HTTP serving. For heavy batch workloads, client/server applications, and Java applications, the server models are the recommended solution, assuming there is a relatively small 5250 interactive workload. The AS/400e 7xx server models combine the best of both worlds. They offer excellent batch and client/server workload performance and highly scalable 5250 capacity from optionally available interactive feature modules.

A key aspect of capacity planning is understanding workload growth by application type. We may have a relatively simple environment where the AS/400 system is dedicated strictly as a Web server, and we expect a 20% compounded growth rate per quarter. Most environments may not be so simple since they have a combination of applications and database requirements. We may have a slight decrease in the interactive workload forecast, but an aggressive increase in transaction-oriented, Web-based applications such as Java servlets.

For our growth scenario, we want to model a 30% growth rate per quarter for the HTTP server load, and a 10% growth rate per quarter for other workloads. We will use the BEST/1 application to help analyze a growth scenario for the Web site, using the measured performance data recorded earlier.

7.3.1.1 Preparing a BEST/1 model from performance data

We will use the existing AS/400 Performance Monitor data as a basis for analyzing future requirements. If you are unfamiliar with the BEST/1 product, refer to *AS/400 Server Capacity Planning*, SG24-2159, or *BEST/1 Capacity Planning*, SC41-5341. Follow these steps:

1. Enter the Start BEST/1 (`STRBEST`) command at the AS/400 command line.
2. Enter option `1` to work with the models.
3. Go to the blank entry fields, and enter `1` to create a model and name it appropriately. Press Enter.
4. Enter option `1` to create a model from the performance data.
5. Enter the library containing the Performance Monitor data. Press F4 to select a representative sample of your workload. Press F18 to sort the data by CPU utilization.
6. Enter `1` to select the appropriate data. See Figure 85 on page 136.

```
                          Select Time Interval

Library  . . . . . :   HTTPPERF      Performance member . . :   Q99228000

Type option, press Enter.  Select first and last interval.
  1=Select

                              ---Transaction---   --CPU Util---   I/Os per Sec
Opt   Date       Time          Count   Rsp Time   Total    Inter   Sync    Async
      08/16/99   16:16:54         34        1.0      37        0     38      329
      08/16/99   16:01:53         65         .2      27        0     78       76
      08/16/99   15:16:49         10         .1      18        0     24      138
 1    08/16/99   10:16:17        173         .0      15        0     40       77
      08/16/99   14:16:42          0         .0      14        0     18      122
      08/16/99   10:31:19          0         .0      13        0     24       40
      08/16/99   15:01:47          0         .0      13        0     22       35
      08/16/99   09:16:12         41         .1      13        0     23       50
      08/16/99   14:31:45          0         .0      12        0     22       41
      08/16/99   11:16:23          0         .0      12        0     19       83
      08/16/99   09:31:13         66         .2      12        0     27       44
```

Figure 85. Select a time interval in BEST/1

7. Enter a name and library for the model. Press Enter.
8. Enter option `2` to classify jobs into workloads. Press Enter.
9. Enter option `3` to choose a job name category. Press Enter.
10. At the Edit Job Classifications display, enter a name for the HTTP server workload in the workload column. Enter `QDEFAULT` in another column (this allows us to categorize work as an HTTP server and all others).
11. Press F9 to view the job performance data. Select the appropriate HTTP server job for your environment (option 1). Next, scroll down the job list and select the IPRTRxxxxx jobs. Enter the name of the HTTP server workload in the Workload field at the top (Figure 86). Press Enter.

```
                          Assign Jobs to Workloads

Workload . . . . . .   HTTPSERVER

Type options, press Enter.  Unassigned jobs become part of workload QDEFAULT.
  1=Assign to above workload   2=Unassign

                               Number of        CPU        I/O
Opt  Workload    Job Name    Transactions     Seconds     Count
                 ETH-ETHLIN           0         .000         0
                 GATE                 0        2.070         0
                 GROWEBUSIN           0        1.431      1213
 1               HTTP80               0     1850.172    161032




 1               IPRTR00016           0       14.997         2
 1               IPRTR00017           0        3.080         0
 1               IPRTR00018           0         .000         0
 1               IPRTR00019           0         .000         0
```

Figure 86. Assign jobs to workloads in BEST/1

12. Press Enter to accept the paging behavior defaults.
13. At the Define Non-Interactive Transactions screen, specify the HTTP server workload activity as *NONE. Enter a transaction rate in hits per hour. In our example, we had about 15,000 hits per hour (Figure 87).

```
                    Define Non-Interactive Transactions

Job classification category  . . . . . :   Job Name


Type choices, press Enter.

               ---Activity Counted as Transaction---     Total Transactions
Workload          Type                  Quantity         when Type = *NONE
QDEFAULT          *LGLIO                  100.0                        0
HTTPSERVER        *NONE                   100.0                    15000
```

Figure 87. Define non-interactive transactions in BEST/1

14. Save the job classification member in an appropriate library.

7.3.1.2 Modeling workload growth scenarios

In the previous section, we created a BEST/1 model based on the Performance Monitor data. We use this data to understand resource requirements for the future workload growth: 30% per quarter for the HTTP workload and 10% for the other workload. Follow this process:

1. At the work with BEST/1 Models menu, enter option 5 next to the model you just created.
2. Enter option 7 to specify a workload growth scenario.
3. Fill in the workload growth parameters as necessary (Figure 88).

```
                    Specify Growth of Workload Activity

Type information, press Enter to analyze model.
  Determine new configuration . . . . . .   Y    Y=Yes, N=No
  Periods to analyze  . . . . . . . . . .   6    1 - 10

  Period 1  . . . . . . . . . . . . . .   F_1999        Name
  Period 2  . . . . . . . . . . . . . .   W_1999        Name
  Period 3  . . . . . . . . . . . . . .   SP_2000       Name
  Period 4  . . . . . . . . . . . . . .   SU_2000       Name
  Period 5  . . . . . . . . . . . . . .   F_2000        Name

            ------Percent Change in Workload Activity-------
Workload    Period 1  Period 2  Period 3  Period 4  Period 5
HTTPSERVER     30.0      30.0      30.0      30.0      30.0
QCMN           20.0      20.0      20.0      20.0      20.0
QDEFAULT       10.0      10.0      10.0      10.0      10.0
```

Figure 88. Specify growth for workloads in BEST/1

4. Press Enter to analyze the model.
5. View the results, such as the analysis report, and look for potential resource utilization problems.

```
                          Display Analysis Summary

                                 Stor      CPU  -Disk IOPs--  -Disk Ctls--  -Disk Arms--
Period     CPU Model             (MB)     Util   Nbr    Util   Nbr    Util   Nbr    Util
F_1999     53S 2157              1536     19.5     2    11.0     8     1.2    36     3.0
W_1999     53S 2157              1536     23.3     2    12.6     8     1.3    36     3.4
SP_2000    53S 2157              1536     28.1     2    14.6     8     1.5    36     3.9
SU_2000    53S 2157              1536     34.0     2    16.9     8     1.8    36     4.6
F_2000     53S 2157              1536     41.5     2    19.7     8     2.1    36     5.4
W_2000     53S 2157              1536     51.0     2    23.2     8     2.4    36     6.3
                                                                                  Bottom
           ----Inter Rsp Time----  -------Inter--------  -----Non-Inter------
Period      Local    LAN     WAN   CPU Util    Trans/Hr  CPU Util    Trans/Hr
F_1999         .0    1.2      .0        .2         267     19.3        22100
W_1999         .0    1.2      .0        .2         294     23.1        28210
SP_2000        .0    1.2      .0        .2         323     27.8        36102
SU_2000        .0    1.2      .0        .2         355     33.8        46303
F_2000         .0    1.2      .0        .3         391     41.3        59503
W_2000         .0    1.3      .0        .3         430     50.7        76592

                                                                          Bottom
```

Figure 89. Display summary information in BEST/1

If you strictly focus on CPU utilization, the 53S server, in this example, will be able to handle the expected load over the next six periods. However, this does not factor in new workloads, such as adding a Java servlet-based transaction application.

7.3.2 Disk arms

Capacity planning for disk storage and disk arm requirements requires careful analysis of two facets: aggregate storage capacity (the number of GB or TB) and disk arm considerations (the number of actual drives). Both can have a significant effect on performance.

The aggregate storage capacity is relatively easy to measure and plan for, whether you periodically use the Performance Monitor data or the AS/400 `WRKDSKSTS` command. As actual disk storage increases past 50% of the overall capacity, service times start to increase, and wait times (queuing) become even more pronounced. In fact, if disk utilization exceeds 90%, the OS/400 operating system gives you a warning. Similarly, the disk IOP utilization should be kept under 60%. Again, wait times (queuing) become more of a factor in the overall response time.

In 6.3.2, “Disk arms” on page 97, we discuss the importance of having enough disk arms to ensure that the processor is not waiting too long for data from the disk. The effect of not enough disk arms is that wait time becomes a relatively large portion of the total response time (wait time + service time). Also, smaller disk drives have less surface area to traverse in finding the appropriate data and are more conducive to parallel processing.

The AS/400 Performance Monitor trace offers a detailed analysis of our disk activity. The system report shows disk storage utilization, IOP utilization, average size of each disk I/O transaction, and a breakdown of response time in terms of service time and wait time. The component report gives even greater detail, such

as accesses by disk sector and read/write cache efficiency. Let's look at portions from the system report (Figure 90) to understand the disk activity in our example.

```
Member  . . . : Q992280000 Model/Serial  . : 53S-2157/10-1D09M   Main storage . . : 1536.0 M   Started  . . . . :
08/16/99 00:00:1
  Library . . : HTTPPERF   System name . . :         RCHAS406   Version/Release  :   4/ 3.0   Stopped  . . . . : 08/16/99
23:42:3
       Unit              Size   IOP    IOP           Dsk CPU  ASP   --Percent--    Op Per    K Per     - Average Time
Per I/O --
 Unit   Name       Type   (M)    Util   Name          Util     ID    Full   Util    Second     I/O      Service   Wait
Response
 ----   ---------- ---- -------  ----   ------------  -------  ---   ----   ----  ---------  ---------  -------  ------
--------
 0028   DD028      6607   3,670   5.8   SI04             .0    01   76.7    .8       3.23       5.0    .0024   .0000
.0024
 0029   DD030      6607   3,670   5.8   SI04             .0    01   77.4    .8       3.26       5.0    .0024   .0000
.0024
 0030   DD036      6607   3,670   5.8   SI04             .0    01   76.2    .8       3.23       5.0    .0024   .0000
.0024
 0031   DD031      6607   3,670   5.8   SI04             .0    01   76.3    .8       3.23       5.0    .0024   .0000
.0024
 0032   DD032      6607   3,670   5.8   SI04             .0    01   76.0    .8       3.37       5.0    .0023   .0000
.0023
 0033   DD033      6607   3,670   5.8   SI04             .0    01   76.3    .8       3.33       5.1    .0024   .0000
.0024
 0034   DD034      6607   3,670   5.8   SI04             .0    01   76.3    .8       3.38       4.9    .0023   .0000
.0023
Total                 134,212
Average                                                        71.0    .5       1.79       7.7    .0027   .0006
```

Figure 90. AS/400 performance system report

As shown in Figure 90, note that the IOP utilization is low, 5.8%, and that the overall disk capacity utilization is 71%. The average disk response time is .033 seconds, with less than 20% comprised of wait time. It appears that, given the total disk size of 134 GB, the 34+ disk arms of 4 GB are an optimal solution. At this point, you should consider increasing the disk capacity, since it is over 70% utilization and you need to allow for growth. Since IOP utilization is low and wait time is a small percentage of total response time, you may consider a few larger disk arms to meet this requirement.

The individual disk utilization rate, in this example, averages less than 1%, and I/O operations average 1.8 per second. Because we recorded approximately 350,000 hits during this period, this works out to about 4 hits per second. Therefore, the disk I/O rate is just under 50% of our overall hit rate. Note that the report also shows that the average request is just under 8 KB.

We can also use the BEST/1 capacity sizing example to look at disk IOP load recommendations, given the expected growth of 30% per quarter for HTTP workload, and 10% for the other workload. See the example in Figure 91.

```
                                Display Analysis Summary

                                  Stor      CPU  -Disk IOPs--  -Disk Ctls--  -Disk Arms--
Period     CPU Model              (MB)     Util   Nbr    Util   Nbr    Util   Nbr    Util
F_1999     53S 2157               1536      8.2     2     4.8     8      .5    36     1.3
W_1999     53S 2157               1536     12.3     2     7.3     8      .8    36     1.9
SP_2000    53S 2157               1536     16.4     2     9.7     8     1.0    36     2.6
SU_2000    53S 2157               1536     20.5     2    12.1     8     1.3    36     3.2
F_2000     53S 2157               1536     25.5     2    15.0     8     1.6    36     4.0
W_2000     53S 2157               1536     31.5     2    18.6     8     2.0    36     5.0
```

Figure 91. Analysis summary in BEST/1

The BEST/1 model predicts the disk IOP utilization, given the indicated workload growth. Note that, even in the W_2000 period, we are well within the guidelines of disk IOP utilization. However, you must manually analyze any disk capacity increase requirements, such as adding Web pages and other files to the site, plus any additional disk requirements for new applications, such as database access.

The data for this analysis shows that disk capacity and arms do not have a significant performance impact, given the low IOP utilization and disk wait time. In our example, we had the advantage of using many small disk drives on our machine. In your environment, this may not always be the case, and you may have to look at additional disk arms and, perhaps, IOPs.

7.3.3 Main memory

In 6.3.3, “Main memory” on page 98, we discuss main memory and its potential impact on overall performance. The key metric is page faults—the rate at which the system has to retrieve data from disk that is not already in memory. Data may be in memory by overt action, such as setting HTTP server local caching, or when the Set Object Access (SETOBJACC) command is used. Data may already be in memory, because the AS/400 Expert Cache mechanism does it for you.

For the HTTP server application environment, there are two AS/400 storage pools of which we need to be aware: *MACHINE and *BASE. We chose not to create a private storage pool specifically for the HTTP server, primarily because we had a large amount of main storage. Besides, rather than do it manually, we decided to let the AS/400 operating system manage the memory for us. As part of the capacity planning process, we may need to revisit this strategy if the *BASE pool has a significant number of page faults.

Similar to our discussion on disk arms and DASD, you may periodically run the Work with System Status (WRKSYSSTS) command on the AS/400 server and look at the page fault data. Or, do a more in-depth analysis using the AS/400 Performance Monitor.

The AS/400 Performance Monitor trace offers a detailed analysis of the storage pool activity. The system report includes database and non-database page fault data, plus statistics on the number of active-to-wait, wait-to-ineligible, and active-to-ineligible job state transitions per minute. The component report gives even greater detail, such as CPU utilization and the page faults per unit of time, so you can determine peak and average values. Let’s look at portions from the system report (Figure 92) to understand the *BASE pool activity, since that is where the user jobs are running. Note that, in practice, one should perform the same analysis on other relevant pools, such as *MACHINE.

```
ASM01 - V04R03M00 - xxxxxx
                                                System Report                                                   9/07/99 15:05:4
                                             Storage Pool Utilization                                             Page 000
                                                System Report
 Member  . . . : Q992280000 Model/Serial  . : 53S-2157/10-1D09M   Main storage . . : 1536.0 M   Started  . . . . : 08/16/99
00:00:1
   Library . . : HTTPPERF   System name . . :         RCHAS406   Version/Release  :   4/ 3.0   Stopped  . . . . : 08/16/99
23:42:3
                                                                  ---------- Avg Per Second ---------  ---- Avg Per Minute
-----
 Pool  Expert      Size      Act    CPU        Number    Average   ------ DB ------   ---- Non-DB ----    Act-     Wait-
Act-
  ID   Cache       (K)      Lvl    Util        Tns      Response    Fault   Pages     Fault   Pages      Wait     Inel    Inel
 ----  ------- -----------  -----  -----     ----------- --------   ------- -------   ------- -------   --------  -------
-------
 *01     0        157,780     0    .6            0       .00        .0      .0       .4      .5        21        0       0
 *02     3      1,399,356   129    7.4         951       .16        .7    28.1      6.4    31.6       982        0       0
  03     0         15,728     6    .0            0       .00        .0      .0       .0      .0         0        0       0
 Total          1,572,864          8.1         951                .8    28.2      6.9    32.2     1,003        0       0
 Average
```

Figure 92. Performance system report showing memory information

Pool ID 01 is *MACHINE, and ID 02 is *BASE. The page faults per second averages are .7 for database and 6.4 for non-database accesses. Generally, the total page faults per second under 10 is acceptable. Adding main memory yields little noticeable improvement. Because a significant amount of HTTP serving is taking place, non-database faulting is higher. Also, notice that the *MACHINE pool has a low fault rate. Having over 1.5 GB of main memory also helps considerably.

This tells us that, on average, page faulting is within acceptable limits today. What about at peak access times? What about the growth scenario? Let's look at the performance component report (Figure 93) to answer the first question.

```
 Pool                                Avg                 ---------- Avg Per Second ----------    ----- Avg Per Minute ----
  Itv      Size       Act      Total        Rsp      CPU      ------ DB ------   ----- Non-DB -----    Act-      Wait-      Act-
  End      (KB)       Level    Tns          Time     Util     Faults    Pages    Faults    Pages      Wait      Inel       Inel
 -----  -----------   -----  -----------    -----    ----     -------   ------   -------   ------    ------    ------     -----
 10:01   1,400,748      129           29      .13     10.5         1.0        2       9.2        52      1200         0
 10:16   1,391,664      129          173      .03     14.5         1.5       44      10.7        58      1580         0
 10:31   1,393,732      129            1      .00     12.7         1.0        2      10.1        59      1279         0
 10:46   1,395,764      129            0      .00      9.9          .7        1       9.9        59      1239         0
 11:01   1,397,784      129            1      .00     10.7          .7        1       9.6        55      1268         0
 11:16   1,399,764      129            0      .00     11.3          .5       83       6.9        39      1194         0
 11:31   1,390,632      129            2      .00      8.1         1.1        1       6.9        38      1072         0
 11:46   1,392,700      129          115      .09      7.9          .7        1       5.7        30      1116         0
 12:01   1,394,756      129           27      .18      7.9          .5        1       5.9        33      1114         0
 12:16   1,396,776      129           66      .13     11.1         1.0       93      10.3        47      1103         0
 12:31   1,398,776      129            1      .00      8.0          .6        1       7.8        43      1082         0
 12:46   1,400,748      129            0      .00      8.3          .9        1       8.8        46      1122         0
 13:01   1,391,668      129            0      .00      7.2          .6        1       9.0        54      1128         0
 13:16   1,393,732      129            0      .00     10.8          .6      110       6.6        39      1054         0
 13:31   1,395,764      129            9      .00      8.0         1.4        2       7.8        44      1125         0
 13:46   1,397,784      129           33      .06     10.4          .7        1       7.3        40      1115         0
 14:01   1,399,764      129            0      .00     10.1          .9        1       9.4        50      1243         0
 14:16   1,390,632      129            0      .00     13.5          .5      136       6.6        37      1165         0
 14:31   1,392,704      129            1      .00     11.8         1.1        2      10.2        52      1272         0
 14:46   1,394,756      129            0      .00     10.4         1.1        1       7.5        42      1182         0
 15:01   1,396,776      129            0      .00     12.4          .9        1       9.5        52      1174         0
 15:16   1,398,776      129           12      .08     17.4          .9      155      10.0        61      1439         0
 15:31   1,400,748      129            1      .00      9.5          .9        1       7.7        43      1093         0
 15:46   1,376,220      129            0      .00     10.9         3.5        6      34.8        74      1184         0
 16:01   1,398,336      129           67      .17     25.8         9.4       26      46.0        84      1136         0
 16:16   1,400,308      129           36     1.02     35.2         2.2      188       8.6        40      1068         0
 16:31   1,391,212      129          155      .06     10.4         1.8       62      10.3        60      1197         0
 16:46   1,393,272      129           59      .59      8.0         1.4        2      11.3        54      1048         0
 17:01   1,395,324      129            0      .00      7.0          .7        1       8.7        44       973         0
 17:16   1,394,072      129            0      .00      4.6          .4        0       4.1        25       860         0
 17:31   1,396,108      129            1      .00      9.3          .2      227       4.6        26       839         0
 17:47   1,398,112      129            0      .00      4.0          .4        0       4.2        23       797         0
 18:02   1,392,244      129            0      .00      4.4          .5        1       3.8        18       815         0
```

*Figure 93. Component report : *BASE storage pool*

Figure 93 on page 141 offers a wealth of important information about the Web site. The average CPU utilization for the site is about 8%. However, notice the variation. Obviously, certain times are busier than others. Note that, from about 15:45 to 16:15, there was a usage spike that affected CPU utilization and page faults for database and non-database access. This could have been one or more large queries, for example, or just a time during the day. Other than this 30 or so minute interval, it appears that the main memory is adequate. Also, it makes no sense to create a private memory pool strictly for the HTTP server tasks.

Since we forecasted a sizeable future growth, we must determine if our memory is adequate. The next step is to use the BEST/1 tool to understand the projected page fault behavior based upon the planned workload increase. At the Work with BEST/1 Model screen, enter option 7 to specify workload growth and analyze the model. After this completes, look at the main storage pool report results. Figure 94 displays a summary of what the model shows.

```
Period:   Period 1
Pool    Act        Size    Ineligible    -----Avg Number-----    Sync
Reads
 ID     Lvl        (KB)    Wait (sec)    Active    Ineligible      per
Sec
  1       0      162106          .0          .0          .0           .3
  2     129     1395030          .0          .4          .0          6.4
  3       6       15728          .0          .0          .0           .0
Period:   Period 2
Pool    Act        Size    Ineligible    -----Avg Number-----    Sync
Reads
 ID     Lvl        (KB)    Wait (sec)    Active    Ineligible      per
Sec
  1       0      162106          .0          .0          .0           .4
  2     129     1395030          .0          .6          .0          9.6
  3       6       15728          .0          .0          .0           .0
.
.
.
Period:   Period 5
Pool    Act        Size    Ineligible    -----Avg Number-----    Sync
Reads
 ID     Lvl        (KB)    Wait (sec)    Active    Ineligible      per
Sec
  1       0      162106          .0          .1          .0          1.1
  2     129     1395030          .0         1.6          .0         24.5
  3       6       15728          .0          .0          .0           .0
```

Figure 94. Display Main Storage Pool report

Note that, as traffic and workload increases, the page faults (Sync Reads per Sec) also increase. The BEST/1 tool considers a page fault of 30 per second acceptable. For Internet connected users with low bandwidth and several second response times, this may be acceptable. LAN connected users may see a small response time increase. As in all of our other analyses, we must mention that this does not factor in any new application workloads. They would have to be factored in addition to these results.

7.3.4 Communications IOP

The last critical component to analyze is the communications IOP and the line utilization. In 6.3.4, “Communications IOPs” on page 100, we discuss three key facets, including bandwidth, IOP utilization, and number of hits per second. Any

of these can be a potential performance problem, so you must analyze each one properly.

We will restrict our coverage to LAN IOPs. However, most of the concepts also apply to WAN IOP resources. Both have bandwidth considerations, IOP utilization percentages, and a finite amount of hits per second capable. We cover bandwidth considerations in more detail in 7.5, “Capacity planning for network resources” on page 146.

First, let’s look at bandwidth. Since the server has to support multiple workloads, HTTP, and other jobs, the HTTP server access log throughput readings will not include everything. Therefore, you need to rely on the AS/400 Performance Monitor data. Figure 95 shows an example of one day’s worth of data.

```
                                    System Report
                                Communications Summary
                                    System Report
Member  . . . : Q992280000 Model/Serial  . : 53S-2157/10-123456   Main storage . . :1536.0 M

  Library . . : HTTPPERF   System name . . :          MY400       Version/Release  : 4/ 3.0

     IOP Name/                        Line      Avg     Max          ------ Bytes Per Second

       Line             Protocol     Speed     Util    Util    Received        Transmitted
--------------------    --------   ---------   ----    ----    ----------      -----------
CC03          (2617)
  ETHLIN002              ELAN/H      10000.0     3       8        4180.9          38118.9
```

Figure 95. Performance system report showing communication data

Note that the line utilization is quite low. However, this is just a one day snapshot. You need to look at additional data, shown in Table 24, before concluding that bandwidth is more than adequate.

Table 24. Number of bytes received and sent

Day	Bytes received/sec	Bytes sent/sec	Total bytes/sec	Bits/sec (bytes * 8)
1	4 KB	38 KB	42 KB	336 KB
2	4 KB	40 KB	44 KB	352 KB
3	3 KB	24 KB	27 KB	216 KB
4	3 KB	26 KB	29 KB	232 KB
5	5 KB	44 KB	49 KB	392 KB
6	5 KB	43 KB	48 KB	384 KB
7	4 KB	37 KB	41 KB	328 KB
8	4 KB	41 KB	45 KB	360 KB

On average, 2% to 4% of the 10 Mbps Ethernet line is used, so bandwidth is not a problem. Next, we look at the component report and IOP utilization (Figure 96 on page 144).

```
Component Report                                    9/09/99 15:20:4
                                                     IOP Utilizations                                                    Page  14
                                                     HTTP Server Test
Member  . . . : Q992280000 Model/Serial  . : 53S-2157/10-1D09M   Main storage . . : 1536.0  M  Started . . . . :  08/16/99 00:00:1
  Library . . : HTTPPERF   System name . . :            AS406   Version/Release  :    4/ 3.0  Stopped . . . . :  08/16/99 23:42:3
                    --- IOP Processor Util ---     DASD      -- KBytes Transmitted --    Available
IOP                 Total   Comm   LWSC   DASD    Ops/Sec       IOP          System       Storage     Util 2
------------------  -----  -----  -----  -----  -----------  -----------  -----------   -----------   ------
CC06          (2623)     .3     .0                                   95           24     1,600,396      .0
SI01          (6513)     .4                                23
SI02          (6512)    4.4                                23
SI03          (6501)     .2                                38
SI04          (6512)    5.8                                38
CC01          (2617)     .6     .0     .0     .0                     95           94     4,827,028      .0
CC02          (2617)     .6     .0     .0     .0                     95           94     4,827,028      .0
CC03          (2617)    12.4   12.4     .0     .0                448,353    3,468,567     2,122,617      .0
```

Figure 96. Performance component report: IOP utilization

Note that the IOP utilization for resource CC03 works out to 12.4%, which is well within the guidelines of less than 40%. Note that it is approximately four times the line utilization. We conclude that the single Ethernet IOP is currently adequate.

The next step deals with the number of hits per second that the communications IOP experiences. Remember from 6.3.4, “Communications IOPs” on page 100, that the LAN IOP cards should be kept to less than 100 hits per second. In 7.2.2, “Categorizing the page objects content” on page 132, we specified that the site had 339,000 hits over a 24-hour period. Doing the math, we see that this is about 4 hits per second, which is well within guidelines.

Lastly, you should look at the BEST/1 capacity plan to determine if your future workload growth will require AS/400 communications resources. At the Work with BEST/1 Model screen, enter option 7 to specify workload growth and analyze the model. After this completes, look at the main storage pool report results. Figure 97 shows a summary of our model.

```
                          Display Comm Resources Report

 Period:   F_1999
                                          Overhead  Rsp Time per  Nbr of  Line Speed
 Resource        Type          Util        Util     Trans (Sec)   Lines  (Kbit/sec)
 CC03            *IOP          18.6
     ETHLIN002   *LINE          5.6         .1          3.26       1      10000.0

 Period:   W_1999
                                          Overhead  Rsp Time per  Nbr of  Line Speed
 Resource        Type          Util        Util     Trans (Sec)   Lines  (Kbit/sec)
 CC03            *IOP          22.3
     ETHLIN002   *LINE          6.7         .1          3.30       1      10000.0
.
.
.
 Period:   F_2000
                                          Overhead  Rsp Time per  Nbr of  Line Speed
 Resource        Type          Util        Util     Trans (Sec)   Lines  (Kbit/sec)
 CC03            *IOP          38.5
     ETHLIN002   *LINE         11.6         .2          3.49       1      10000.0
```

Figure 97. Performance communication resource report

Note that we are approaching the maximum recommended IOP utilization. In fact, in the recommendations section of the BEST/1 analysis, we see the following message, because the last period in our model showed additional resources created:

```
*****  W_2000 Initial Exceptions  *****
 Communications IOP CC03 utilization of 46.20 exceeds objective of 45.00

*****  W_2000 Configuration Changes  *****
 1 2617 IOP(s) created
 1 communications line(s) created with line speed of 10000.0
```

In summary, it appears that there are adequate communications IOP resources for the next several quarters. As with AS/400 CPU, disk arms, and main memory capacity planning mentioned earlier, this only assumes the indicated growth in existing workloads. If you plan to have any new applications, you would have to factor that in separately. This may mean that you need to plan for an additional IOP for your AS/400 Web server earlier than the BEST/1 model indicates.

7.4 Capacity planning for client resources

Do you need to perform capacity planning for the browser client environment, even though you have no control over Internet users? The answer is yes, especially since it is what customers actually see. The network computing, "thin client" model, has had an impact on the size of workstation needed for commercial business application deployment. However, even these workstations can be underpowered. Granted, your customers and business partners (and maybe even your employees) accessing your Internet, intranet, or extranet site may have their own workstation and browser combination that you may not be able to influence. Nonetheless, you should monitor your Web application traffic by periodically activating the HTTP server agent log to determine the browser type and version. You will want answers to such questions as these:

- What types of browsers are being used ... any non-PC browsers, such as cellular telephones?
- What versions of browsers are being used ... are they mostly Version 3 and above ... Version 4 and above?
- Are most, or all, site visitors able to view our JavaScript or style sheet enhanced pages?
- Do my users have SSL-enabled browsers for doing secure transactions?

Here is an example of an entry in an AS/400 HTTP server agent log for a Netscape Version 4.07 user:

```
[02/Nov/1998:12:51:44 +0000] "Mozilla/4.07 [en] (Win95; I)"
```

7.4.1 Intranet workstation considerations

In this environment, you likely provided the end users with PC or other workstation hardware. For graphically rich pages, or those with significant amounts of JavaScript, you may need at least a 100 MHZ processor and 32 MB of RAM in the workstation. You should double these requirements if deploying Java applets on the client. You should also ensure that the network card and all other hardware drivers are recent. Obviously, the greater the amount of non-browser workload on the system, the more hardware resources you need.

Even today's personal computers with lots of RAM and CPU MHZ, plus a 100 Mbps LAN card, will strain if they are running too many heavyweight applications such as graphics editing or full motion video multimedia to the desktop.

From a software perspective, you should attempt to provide your users with as recent an operating system as possible, and keep up with the various service packs. Newer versions of Web browsers support persistent HTTP connections and other performance enhancements that are part of HTTP 1.1, such as byte range file serving. Newer browser versions also generally have better Java performance. You should understand that the internal use of security measures, such as digital certificates and secure sockets communication or client token authentication solutions, add to response time.

7.4.2 Internet and extranet workstation considerations

In this environment, you have little or no control over the workstation hardware, software, and network connection means. You should analyze your agent log to determine if most of your users are using a JavaScript-enabled browser (for example, Netscape Version 2 and later, Microsoft Internet Explorer Version 3 and later). This enables the client code to perform tasks like simple form field validation, which is much faster and less resource intensive than having a CGI or other server-based application do it. It can also be used to determine Java support (or lack thereof) and steer the user to a specific set of directories, based on their browser version. If you are using Java applets, keep in mind that the user's workstation will have to load the Java Virtual Machine and wait for the .class, .jar, and .cab files to be downloaded and processed.

The AS/400 HTTP server on V4R3 and later supports automatic browser detection and enables a specific page to be served to your user, based on the user agent passed to the server by the browser. For example, you may have a "site" for Version 4 or later browsers and another site for earlier browser versions. There are advantages in enabling your server to dynamically determine what page or directory to use based on the browser type. However, as you can guess, this consumes server resources and may not even be necessary in your environment.

If dial-up network users, or users in various places of the universe with low bandwidth capabilities, are accessing your site, the graphical richness of your site will have a moderate to extensive impact on response time. A rule of thumb for the Internet is that a user will only wait 8 seconds to determine if a page is worth downloading or if they click through or load another page.

You also need to be judicious in the use of various security measures. Digital certificates and SSL can add measurably to the client load. If you use a Virtual Private Network (VPN) to connect remote users to your corporate network over the Internet, this, too, will have a modest to extensive impact at the client.

7.5 Capacity planning for network resources

We saw a number of response time examples in Chapter 6, “Sizing Web-based applications” on page 91, that illustrates the high complexity and variability associated with the network and communications infrastructure. We also discussed the virtues of intelligent network design. One of the realities of the networked world is that any bandwidth or network capacity increase deployed is

consumed instantly. Another reality is that your network infrastructure must be resilient enough to handle a variety of applications and traffic 24 hours a day, 7 days a week.

7.5.1 General network considerations and intranet considerations

Regardless of whether your environment is directly or indirectly connected to the Internet, sound business and information technology management practice dictates some level of network infrastructure monitoring. This is absolutely necessary for security purposes, and often accounting reasons (many organizations charge out network costs to organizations based on resource usage). It is also essential for providing appropriate tools to the systems and network administration staff. This can be provided in one or more of the following ways:

- System management frameworks, such as the Tivoli suite of products, which provide equipment inventory, software distribution, network management, application management, and an extensible framework for third party customization and value add applications
- Network management products based on Simple Network Management Protocol (SNMP), SNMP Remote Monitoring (RMON), and device agents or MIBs
- Application management products that rely on active or passive agents that monitor response time, such as VitalSuite from International Network Services, NextPoint from NextPoint Networks, or Pegasus from Ganymede Software
- Single user PC products, such as WhatsUp from IPSwitch, Net.Medic from International Network Services, or Distributed Observer from Network Instruments

7.5.1.1 Reduce or eliminate unnecessary traffic

The first step in network capacity planning is to look for inappropriate or excessive resource usage. Many organizations have one or more connections to a network provider that services their internal Internet requirements and external Internet site on the same line. The networked world and Internet have opened up a wealth of information to your users. However, it also subjects your infrastructure to abuse and other problems, such as:

- Are users accessing inappropriate Web sites or sending or receiving inappropriate e-mail?
- Do response times to the external Web site slow down over the lunch hour when employees do their Internet surfing?
- Are certain organizations or individuals using a disproportionate amount of network resources for Web surfing, file downloads, or large e-mail attachments?
- Do the firewall or proxy server logs show a sizeable amount of Multipurpose Internet Mail Extensions (MIME) content that indicate network resources are being consumed by news feeds, listening to Internet radio stations, or other multimedia broadcasts?
- Is the firewall or proxy server underpowered and causing poor response time?

Certainly, your organization's management will ask questions such as these prior to authorizing significant expenditures for network infrastructure, so you need to look first at ensuring that unnecessary network usage is minimized. Allocating the organization's network infrastructure costs by usage and consumption can have an amazing impact on controlling unnecessary usage.

Network and application policy management solutions can be used to give network and bandwidth priority to critical applications (such as the external Web site) over internal Web surfing. However, these can be complex and are no substitute for having established guidelines on acceptable usage of information processing resources in the organization.

7.5.1.2 Understanding the network traffic patterns

In the host centric application computing model, such as an interactive 5250 terminal application, network traffic is fairly predictable and the 80/20 rule applies. This means that 80% of the traffic is with hosts outside the workgroup and only 20% of the traffic is with hosts within the workgroup. In the peer-to-peer client/server computing model, it is often the opposite. That is, 80% of the network traffic is within the workgroup and 20% of the traffic is outside the workgroup with other hosts. The network computing model puts this somewhere in between and can be difficult to analyze from intuition alone. Your environment may have Web-oriented applications, interactive 5250 applications, e-mail, and other client/server applications. Each have their own network utilization impacts. This highlights the need to understand traffic patterns and plan appropriately. Figure 98 shows the traffic pattern over two years.

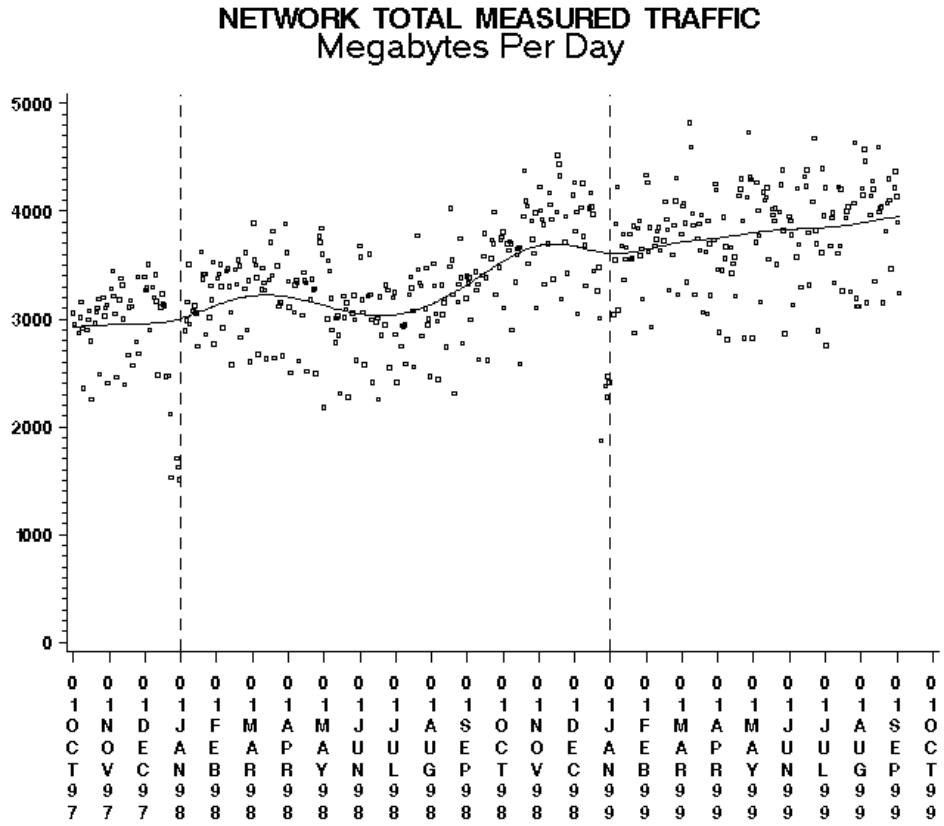


Figure 98. Example of aggregate network usage over time

Most networking hardware infrastructure, such as routers, have an SNMP agent that allows commercial system and network management products to monitor the bandwidth usage and packets forwarded. These can be quite helpful in determining network bottlenecks. For example, you may have multiple servers in a particular subnetwork that often generate excessively high network traffic. Or, your network equipment may be a bit old, and you need to put in a more modern solution.

Figure 99 on page 150 shows an example of how you can periodically assess your network hardware and determine if your infrastructure needs an update or some redesign.

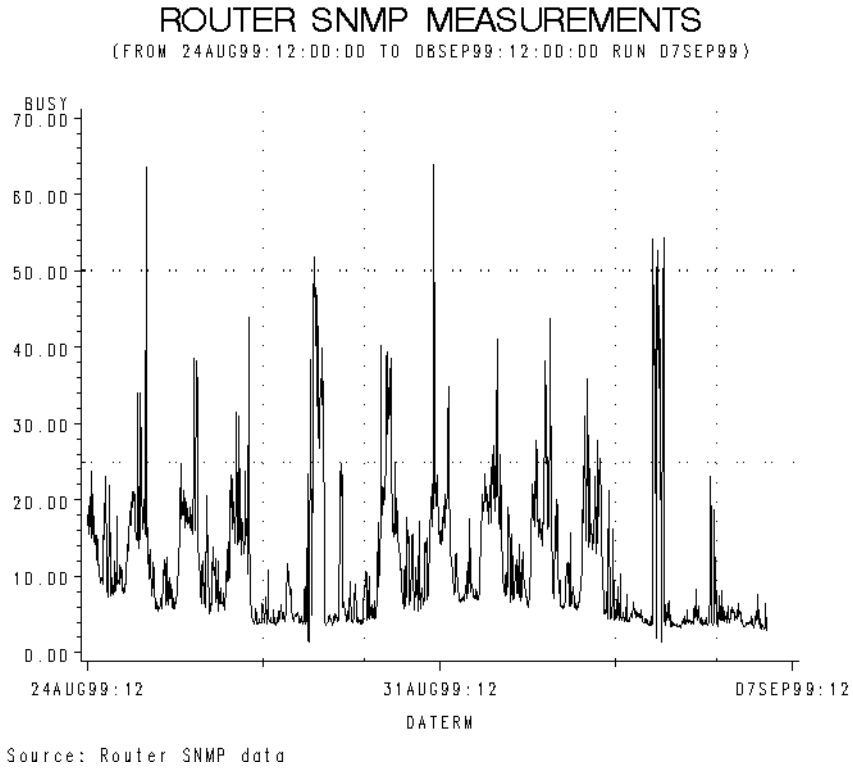


Figure 99. Example of router utilization over time

The analysis of network backbone resources may require special analysis and considerations, particularly for environments with heavy traffic between subnetworks or multi-protocol deployments. Standard LAN resources used on workstations, servers, and routers (16 Mbps Token-Ring or 10 Mbps Ethernet) are usually inadequate for the backbone that connects bridges, routers, switches and other gateways. Solutions, such as FDDI, 100 Mbps Ethernet, or 1000 Mbps Ethernet, are more dependable.

7.5.1.3 Network design and general recommendations

In Chapter 6, “Sizing Web-based applications” on page 91, we discuss the concept of how the network design may have a greater role in overall Web application response time than just raw bandwidth on LAN and WAN connections.

Another consideration is traffic over wide area connections, such as between headquarters and branch offices. Many organizations use a hub and spoke network architecture, which is usually much less expensive than a total peer-to-peer network, especially if you are using leased lines or Frame Relay. However, it introduces a single point of failure and a potentially severe network bottleneck at the hub (Figure 100). You may want to rethink this strategy.

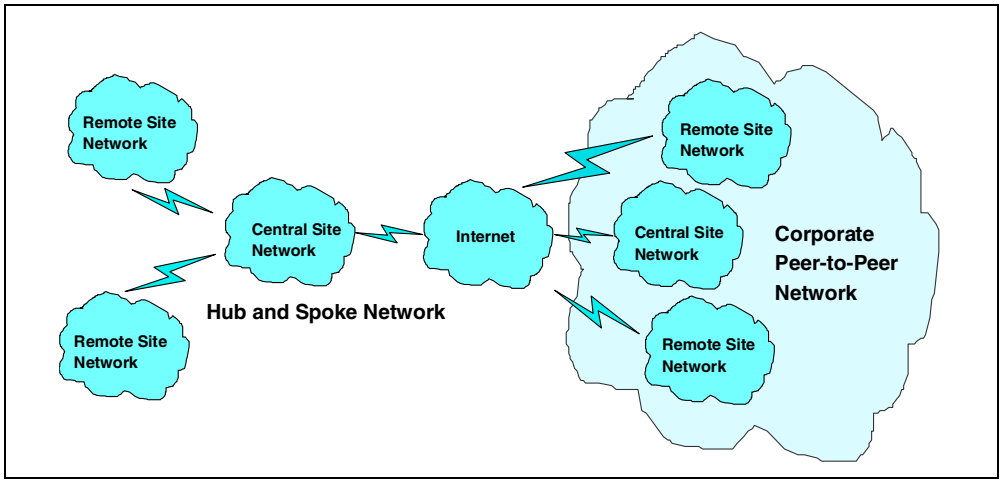


Figure 100. Hub and spoke versus peer-to-peer network

Quite often, the first network capacity problems are at the server and workgroup level, not at the network backbone. For highly loaded network segments, you may need to redesign the network to reduce the network collision domains and provide better utilization of the existing network (Figure 101).

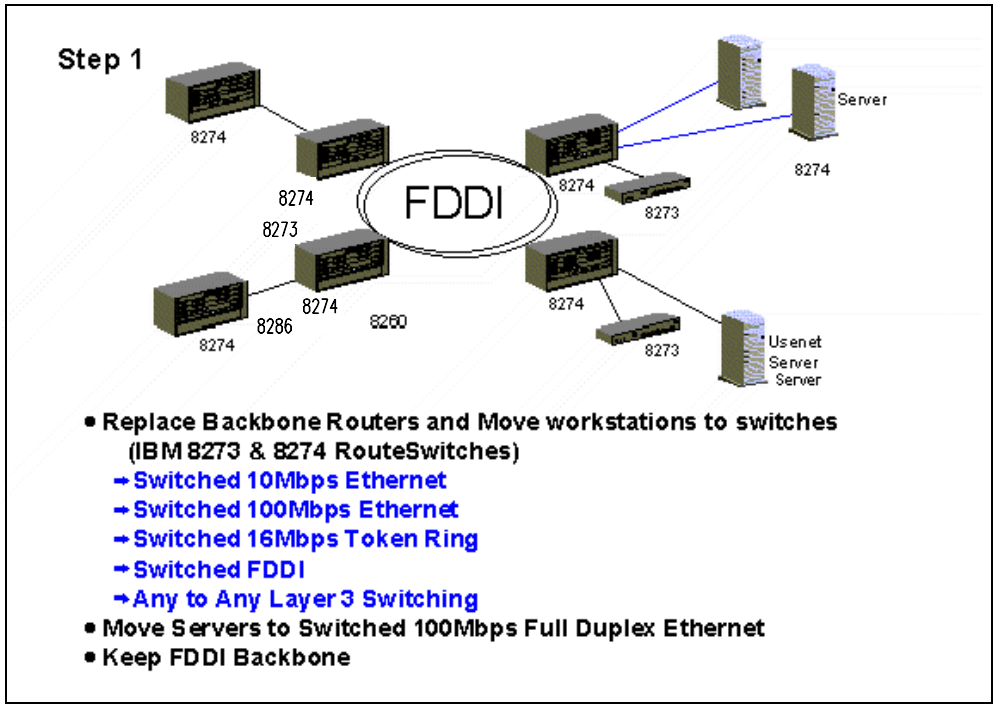


Figure 101. Step 1: Better bandwidth utilization for servers and workgroups

Eventually, you may need to increase the bandwidth on the network backbone to greater than 100 Mbps. Two common solutions are to use a Gigabit Ethernet or ATM.

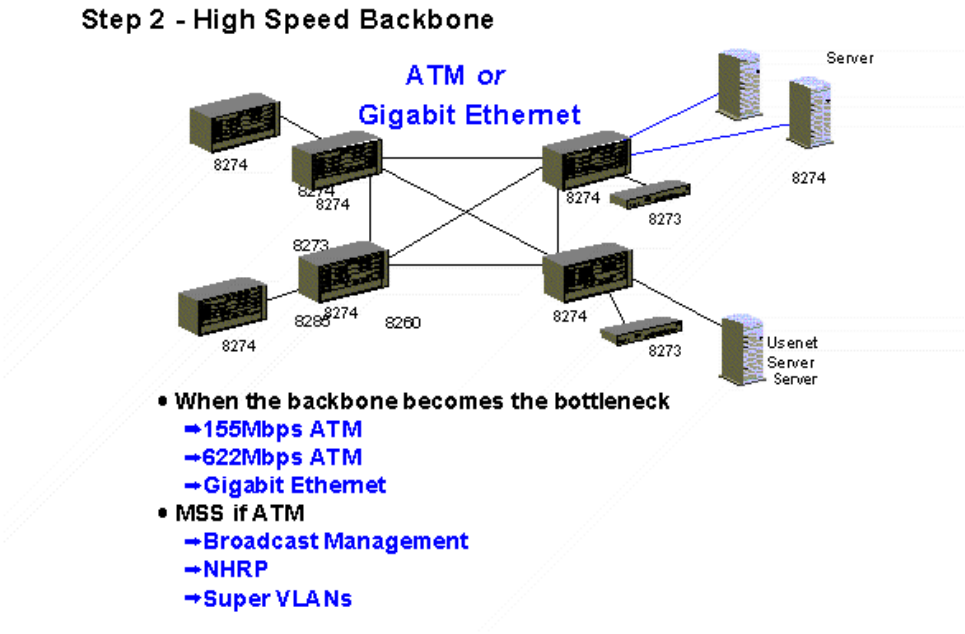


Figure 102. Step 2: Increase network backbone capacity

For heavily loaded intranet segments and servers, consider one or more of the following options:

- Reduce the number of network protocols and move to an all IP network to simplify management.
- For multiple servers on a subnetwork, use LAN switches to separate one or more servers into their own isolated collision domain.
- Put heavily utilized servers, especially proxy or SOCKS servers, on a dedicated LAN switch port and set their network interface card to full duplex operation.
- For instances of heavy traffic between physically separate LAN segments, use a LAN switch to put these segments into the same virtual LAN.
- Use HTTP proxy servers to place Web site content closer to the actual end users.
- Consider investments in increasing bandwidth, such as 100 or 1000 Mbps Ethernet in the network backbone first, servers second, and individual workstations last. High bandwidth at the client is useless unless the rest of the infrastructure can keep up with it.

You also need to forecast potential increases in network load. Some of the networking technologies trade press publications put the yearly bandwidth growth figure at 40%. However, that may not be realistic in your environment. There are many new technologies and application models that are heavy bandwidth consumers. For example, IP-based telephony and other multimedia over data lines can be financially attractive at first glance, but require a significant investment to realize the benefits.

7.5.2 Internet and extranet considerations

Even if you choose to outsource your Web site and Internet mail infrastructure, your organization may still have substantial outbound traffic to the Internet. If you host your own Internet Web site and e-mail support, obviously you need to plan for this network traffic as well. Most organizations need to use the same network infrastructure for internally initiated requests (employees accessing the Web) and externally initiated requests (visitors to the Web site). Each of these environments need to be monitored and have their own unique capacity planning implications. However, you probably can't do much about Internet bandwidth and capacity (or lack thereof). If you haven't experienced it already, you will find that WAN infrastructure and service ranges from readily available 64 Kbps leased or Frame Relay lines, to hard to come by 622 Mbps OC-12 ATM or SONET lines. Cost equivalents in US dollars range from hundreds to hundreds of thousands of dollars per month.

7.5.2.1 Internally initiated Internet traffic considerations

Many organizations today, whether they are schools, medical facilities, manufacturers, or financial institutions, have some level of Internet access. They may also have enabled their users to exchange mail with other users over the Internet. Even rural areas and developing nations see the advantages of being connected to the Internet and are moving this way. Access to the Internet and ongoing growth is inevitable. However, it must be properly managed and budgeted for. This traffic can be monitored from network infrastructure, such as routers, or security infrastructure, such as firewalls and proxy servers. We can analyze proxy server logs with the same tools used to analyze HTTP server access logs. They can help answer such questions as:

- What percentage of wide area bandwidth is being used ...10%, 30% or 90%?
- Is Internet access at the office slower than at home where I use a 28.8 Kbps modem?
- Is my internally generated traffic fairly evenly spread over Monday through Friday, 6 a.m. to 6 p.m., or are there significant spikes at lunchtime or Monday mornings and Friday afternoons?
- Do we have heavily accessed sites of our suppliers or trading partners? Would a proxy caching server help conserve bandwidth and improve response times?
- If I am using a proxy caching server, what percentage of the requests does it service compared to my proxy server ... 50/50, 70/30, 10/90?

This is an example of an AS/400 HTTP server proxy access log entry:

```
9.5.99.118 - - [03/Nov/1998:15:34:11 +0000] "GET http://www.ibm.com/ HTTP/1.0" 200 371
```

This is an example of an AS/400 HTTP server proxy cache access log entry:

```
9.5.99.118 - - [04/Nov/1998:10:58:26 +0000] "GET http://www.as400.ibm.com/ HTTP/1.0" 200 4182
```

For Internet mail usage, which can be substantial, the Lotus Notes R5 Administrator client and Domino R5 server provide outstanding mail usage monitoring tools and are highly recommended. For Web surfing and FTP traffic analysis, you need to rely on logging provided by the routers or proxy servers. If you deploy a proxy caching capable server, such as AS/400 HTTP server on V4R3 or later releases, you can analyze all these questions, other than whether your Internet access is faster at home. The AS/400 proxy server can also support

and log FTP access, which can be helpful since file downloads can consume enormous amounts of bandwidth. For example, we enabled proxy and caching logging on our server, set our browser to use the AS/400 system as our proxy, and did a bit of Web surfing. Some relevant results are shown in Table 25 and Table 26.

Table 25. Proxy server activity log

Summary of Activity by Time Increment				
Time Interval	Hits	Page Views	KBytes Transferred	User Sessions
08:00:00 - 08:59:59	175	22	855 K	1
09:00:00 - 09:59:59	861	431	3,755 K	0
10:00:00 - 10:59:59	3,840	528	23,446 K	0
11:00:00 - 11:59:59	1,461	370	17,719 K	1
12:00:00 - 12:59:59	1,514	401	18,886 K	0
13:00:00 - 13:59:59	0	0	0 K	0
14:00:00 - 14:59:59	1	0	30 K	1
Total	**7,852**	**1,752**	**64,691 K**	**3**

Table 26. Proxy server cache activity log

Summary of Activity by Time Increment				
Time Interval	Hits	Page Views	KBytes Transferred	User Sessions
08:00:00 - 08:59:59	0	0	0 K	0
09:00:00 - 09:59:59	0	0	0 K	0
10:00:00 - 10:59:59	0	0	0 K	0
11:00:00 - 11:59:59	4,169	274	16,444 K	2
12:00:00 - 12:59:59	4,754	282	17,844 K	0
Total	**8,923**	**556**	**34,288 K**	**2**

These tables offer a wealth of information in understanding traffic. For example, about 50% of the Web page requests are serviced by the internal proxy cache and about 50% go through the proxy to the external site. You can also use this information to calculate bandwidth utilization. For example, our peak traffic was during the noon hour. We had 19 MB worth of throughput over a 60 minute interval, which works out to 42 Mbps (from the proxy log only, since the proxy cache was from the local server). If we have a T1 line rated at 1.5 Mbps, this represents 3% line utilization. If we had no proxy cache server, we would have had 36 MB of data, and our line utilization would be about 5%.

```
19 MB/hr / (3600 seconds) * 8 bits/byte  / 1.54 Mbps = 3%
```

You can use this information to determine today's bandwidth and network resources utilizations, and extrapolate it to plan future requirements. For example, we determined that an investment in proxy caching is a good idea in our environment, and we should use it to its full advantage.

7.5.2.2 Externally initiated Internet traffic considerations

Some organizations outsource their Web sites. Yet there are many advantages to hosting your own Internet Web site and mail infrastructure. More often than not, organizations that provide their own Internet Web site and e-mail support structure use the same communications infrastructure for this traffic and their own internal users' Web surfing, file download, and e-mail activities. Therefore, capacity planning takes on additional importance. Since your external Web site represents your organization, you want it to perform well. You need to ensure that your Internet communications infrastructure is managed and planned for properly. You want good responsiveness to your Web site visitors, and your own employees. Ideally, you can provide separate network infrastructures for your

external Web site and your own internal Internet needs. However, business and financial realities will often dictate that this is not possible and that you have to manage both environments over the same network connections.

Certainly, the first order of business is to enable HTTP server access logging and regularly assess your results to get a good, thorough understanding of the traffic and patterns. Ask yourself questions such as these (if you don't, your management will):

- Are hits mostly during our country's business hours or spread across all times of the day and weekends?
- Do we get high access rates to our site immediately after specific events such as product announcements, write-ups in the trade press, or earnings reports?
- Does the proverbial 80/20 rule apply? Are 80% or more of our hits going to a handful of select pages?
- Is our site getting mostly Web surfers that just look at our home page and maybe our product overview at most, or are we also getting serious prospects that want more information or to order products from us?
- When accessing our own Web site from a dial-up connection, as a prospective customer may, does our home page show enough after 5 to 8 seconds to encourage the user to stay at your site?
- Will our site be able handle a 30% traffic increase during the holiday shopping season?
- Is our site getting hit much by hackers or competitors—Telnet or FTP requests from user ID QSECOFR or QUSER, SMTP mail to postmaster, REXEC commands, or strange looking URLs trying to run CGI-BIN programs at our site?

The HTTP server access logs and routers connecting the site to the Internet can be used to help answer these questions and plan for the future. You should analyze your router logs regularly to look for security-oriented incidents that can reduce your network throughput. For example, many commercial sites do not allow ICMP traffic in or out to prevent the "ping of death" and to reduce unnecessary traffic. Also, routers are often configured to pass only certain inbound and outbound protocols (allow HTTP, disallow REXEC, FINGER, etc.). Additionally, routers are often configured to pass certain IP traffic only between designated IP addresses, such as a remote office or trusted supplier. The more filters in place on your routers, the more latency and CPU resource consumed. Newer equipment with faster processors and application-specific integrated circuits can help minimize this effect. However, the realities of the Internet connected world dictate that your peripheral network equipment is the first line of security for the organization. You should also periodically analyze the data transfer and packets per second throughput, as shown in 7.5.1.2, “Understanding the network traffic patterns” on page 148.

The AS/400 HTTP server access log should be periodically analyzed to understand the overall health of your site and plan for future applications and growth (Table 27 on page 156). Earlier chapters showed you a number of tools that can be used to determine overall throughput, the most popular pages, how

the hits are distributed over time, where they are the coming from, etc. We direct our focus specifically to the network implications.

Table 27. AS/400 HTTP server access log results

Summary of Activity by Time Increment				
Time Interval	Hits	Page Views	KBytes Transferred	User Sessions
08/10	338,769	124,927	3,694,576 K	17,907
08/11	283,601	91,661	2,956,030 K	16,911
08/12	326,391	116,526	3,471,072 K	17,369
08/13	289,334	107,600	2,781,932 K	14,959
08/14	87,992	26,833	704,029 K	4,621
08/15	60,274	20,614	546,709 K	4,484
08/16	324,187	116,974	3,307,981 K	17,218
08/17	301,025	127,073	3,0701,766 K	15,044
Total	**2,011,373**	**732,208**	**20,533,095 K**	**108,513**

You can also use the AS/400 Performance Monitor Data to assist with network analysis and planning. In this example, we look at data for Monday, 8/16.

```
                                        System Report
                                                Communications Summary
                                                   System Report
Member  . . . : Q992280000 Model/Serial  . : 53S-2157/10-123456   Main storage . . :1536.0 M

  Library . . : HTTPPERF   System name . . :          MY400       Version/Release  : 4/ 3.0

     IOP Name/                          Line      Avg     Max          ------ Bytes Per Second

       Line             Protocol      Speed     Util    Util    Received        Transmitted
--------------------    --------   ---------    ----    ----    ----------      -----------
CC03          (2617)
  ETHLIN002               ELAN/H      10000.0     3       8        4180.9          38118.9
```

Figure 103. System Report summary

The system report (Figure 103) shows us that of the total bytes sent and received, almost 90% was transmitted and only about 10% was received. You can also use this information to calculate bandwidth utilization. During this 24-hour interval, we averaged 42.3 KB worth of throughput, which works out to .34 Mbps. If we have a T1 line rated at 1.5 Mbps per second, this represents 22% line utilization.

```
42.3 kbytes/second * 8 bits/byte  / 1.54 Mbps = 22%
```

Keep in mind that this is only for externally initiated requests. You may also need to add in the bandwidth consumption of internal users, if appropriate (which we calculated at 3% to 5% in an earlier example). In addition, you must factor in any other WAN or Internet traffic requirements, such as FTP or SMTP mail applications.

7.5.2.3 Network connectivity considerations

You can use this information to determine today's bandwidth and network resources utilizations, and extrapolate it to plan future requirements. For example, it may be a good investment to add another Internet connection with another network service provider. This provides extra bandwidth and load balancing capability, plus failover in case one carrier has a network problem.

OS/400 V4R2 and later releases support this outbound load balancing quite nicely. They allow you to define each ISP connection as a default route. In the example shown in Figure 104 and Figure 105, we define two default routes to the Internet for our one AS/400 network interface card.

```
                        Add TCP/IP Route (ADDTCPRTE)

Type choices, press Enter.

Route destination  . . . . . > *DFTROUTE
Subnet mask  . . . . . > *NONE
Type of service  . . . . . *NORMAL       *MINDELAY, *MAXTHRPUT...
Next hop . . . . . > '205.205.205.1'
Preferred binding interface  . .  205.205.205.18
Maximum transmission unit  . . .  *IFC          576-16388, *IFC
Route metric . . . . . .  1            1-16
Route redistribution . . . . . .  *NO          *NO, *YES
Duplicate route priority . . . .  5            1-10
```

Figure 104. Default route #1

```
                        Add TCP/IP Route (ADDTCPRTE)

Type choices, press Enter.

Route destination  . . . . . > *DFTROUTE
Subnet mask  . . . . . > *NONE
Type of service  . . . . . *NORMAL       *MINDELAY, *MAXTHRPUT...
Next hop . . . . . > '206.206.206.1'
Preferred binding interface  . .  206.206.206.18
Maximum transmission unit  . . .  *IFC          576-16388, *IFC
Route metric . . . . . .  1            1-16
Route redistribution . . . . . .  *NO          *NO, *YES
Duplicate route priority . . . .  5            1-10
```

Figure 105. Default route #2

These routes have equal priority. However, we could have modified the duplicate route priority parameter to give one a preference over the other.

Your network provider may provide you with a fixed bandwidth and fixed fee solution, such as a T1 or fractional T1 line. This is fine for fairly predictable or constant bandwidth requirements. But what about for highly variable or unpredictable network traffic? You have several options, each with their own advantages and disadvantages:

- Purchase additional fixed lines or bandwidth from the carrier (may be the only option).
- Determine if a Frame Relay line with a higher aggregate throughput and modest Committed Information Rate would be appropriate.

- Determine if your network provider can provide a "bandwidth on demand" type solution.
- Consider a collocation arrangement where you place your server on the network provider's premises and pay based upon a fixed amount plus a variable amount for traffic that exceeds a certain threshold.

7.6 Capacity planning for security features

In Chapter 6, “Sizing Web-based applications” on page 91, we perform a number of fairly detailed sizing analyses related to two key security-oriented environments, such as SSL and proxy servers. We learned that security features can have a moderate to major impact on response time and resource usage. If a server is experiencing a greater than expected amount of SSL transactions, it can greatly affect capacity planning. Similarly, as a proxy server’s utilization increases, the queue time increases accordingly. This certainly affects our users’ response time.

7.6.1 SSL environment

For capacity planning in an SSL environment, you need to answer two basic questions. First, how many hits, or what percentage of your traffic used SSL, and how does that compare to your plan? Secondly, what is the impact on server resources?

To answer the first question, look at the access log data, determine the number of secure transactions and amount of SSL-secured data transferred, and compare it to our plan. On our Web site, all of our SSL-secured data resides in the accounts directory. Table 28 shows an example of the most accessed directories.

Table 28. Breakdown of HTTP server traffic by directory

Most Accessed Directories						
	Path to Directory	Hits	% of Total Hits	Non Cached %	Non Cached K Xferred	User Sessions
1	http://rhasm20/home	23,793	51.45%	100%	215,405	5643
2	http://rhasm20/	11,130	24.07%	100%	101,225	2321
3	http://rhasm20/cgi-bin	6,130	13.25%	100%	71,225	1189
4	http://rhasm20/accounts	5,196	11.23%	100%	82,227	1068

You can use the HTTP server access log analysis tools to get an idea of the magnitude of the secure transactions. In our example, you can see that SSL transactions (the accounts directory) account for about 11% of the total hits and about 17% of the data transferred. You can compare this data with our planning data to make any necessary modifications for planning the extra resources for SSL.

The second question is much more difficult to answer. In Chapter 6, we discuss sizing based on application characteristics and the SSL environment. For SSL 40-bit encryption, the server CPU resource consumption was 2.3 to 2.5 times that of static pages. For simple Net.Data SQL pages, for SSL 40-bit encryption, the server CPU resource consumption was 1.1 to 1.3 times that of non-secure Net.Data SQL pages. To complicate matters, your actual application may have its own multiplier effect.

If your site's SSL traffic is comprised completely of static pages, you could estimate the effect of SSL on a V4R3 HTTP server as follows:

```
hits uplift: 11% SSL pages * 2.3 relative CPU usage = 25%
data uplift: 17% SSL traffic * 2.3 relative CPU usage = 39%
```

We should use the data uplift factor for our site, since the log reports show that hits from our SSL-enabled directory generate larger than average data transfers.

If the SSL traffic is from dynamically generated pages, such as CGI programs or Net.Data, the relative uplift for SSL is less. However, the dynamic pages relative CPU usage uplift compared to static pages is substantial. You can go through similar calculations to determine the appropriate uplift for these pages.

If you deploy SSL and dynamically generated pages on your site, capacity planning can range from moderately to extremely difficult.

7.6.2 Proxy environment

We discussed using your AS/400 system as an HTTP proxy, and an HTTP proxy cache server. In the proxy-only environment, the AS/400 system acts mostly as a communications gateway between the browser client and the server. If you are using the server as a proxy cache, the workload is much more substantial. However, it is generally worth the investment because it can give end users a better response time.

Many proxy server implementations use one or more dedicated machines for this task. From an AS/400 perspective, you don't need another machine, but you may want to create a separate HTTP server instance dedicated strictly for the proxy function for several reasons:

- The proxy or proxy cache function does not consume resources on the same server instance that our application server is using.
- The logging and other system administration tasks associated with the proxy server are usually different from those of a Web application server.
- The proxy server can be reconfigured, or stopped and started, without impacting the Web application server.
- You may want to restrict access to the proxy server by placing protection directives in the HTTP server configuration file that forces users to provide an authorized user ID and password, or restrict access by IP address.

If you deploy an AS/400 proxy server, you can use the standard performance monitor reports to determine resource utilization. Communications IOP, CPU, and possibly main memory may be most impacted. Similarly, you can use the BEST/1 modeling tool to forecast resource requirements for your expected traffic growth.

If the server is acting as a caching proxy, the analysis is more complicated. In addition to the workload from acting as an application gateway, there are additional processes that accept the client's Web page request and determine if any of the objects are stored locally at the server. If so, the proxy cache acts as the Web server. For objects not on the server, the proxy function is invoked to retrieve the objects on behalf of the client. The actual mechanisms involved with proxy caching involve much more because many additional mandatory and optional tasks may be involved:

- Determining if the cached object is current
- Caching the content from some Web sites but not others
- Bypassing the proxy server for certain Web sites or networks, such as intranet environments
- Managing the cache within the allotted storage amount and number of objects
- Participating in a proxy chaining environment where you have additional proxy servers to the Internet

In Chapter 6, “Sizing Web-based applications” on page 91, we perform a moderately extensive exercise to assist in sizing the proxy caching load on the AS/400 Web server. As in other analyses, we rely on Web server log files, plus the AS/400 Performance Monitor, to measure resource utilization and to correlate the transaction rates with the measured results. You can use the data in conjunction with the BEST/1 sizing tool for capacity planning.

We use the data obtained in our Chapter 6 proxy cache example and in 7.5.2.1, “Internally initiated Internet traffic considerations” on page 153. To summarize, these are the important parameters obtained:

- In total, the proxy cache server serviced an average of .4 page hits/second, 3.2 object hits/second, and 153 Kbps.
- The proxy cache function serviced an average of 42% of the total page hits, 75% of the total object hits, and 48% of the overall traffic.
- The workload comprised approximately 1.3% CPU utilization.

Since these tests were run on a fairly small scale, we need to understand the overall impact to the CPU, communications IOP, and other resources in a more heavily loaded "real world" environment. However, even if we had more real world data, our methodology is exactly the same. We use these data in conjunction with the BEST/1 capacity planning model.

In this example, we use AS/400 Performance Monitor data obtained while the proxy cache server was running. We create the model similar to our previous analyses. We also specify 5,000 transactions per hour as a basis (just under 14 object hits per second, or four times our original load). We run a BEST/1 capacity planning model with the inputs shown in Figure 106.

```
                    Specify Growth of Workload Activity

Type information, press Enter to analyze model.
  Determine new configuration . . . . . .   Y    Y=Yes, N=No
  Periods to analyze  . . . . . .           10   1 - 10

  Period 1  . . . . . .   *current       Name
  Period 2  . . . . . .   Period 1       Name
  Period 3  . . . . . .   Period 2       Name
  Period 4  . . . . . .   Period 3       Name
  Period 5  . . . . . .   Period 4       Name

            ------Percent Change in Workload Activity-------
Workload    Period 1  Period 2  Period 3  Period 4  Period 5
HTTPSERVER       .0     100.0     100.0      50.0      20.0
QCMN             .0      20.0      20.0      20.0      20.0
QDEFAULT         .0      20.0      20.0      20.0      20.0
```

Figure 106. Growth o f workloads in BEST/1

A summary, based on the BEST/1 Analysis report is shown in Figure 107.

```
                          Display Analysis Summary

                          Stor     CPU  -Disk IOPs--  -Disk Ctls--  -Disk Arms--
Period    CPU Model       (MB)    Util   Nbr    Util   Nbr    Util   Nbr    Util
*current  170 2388        2816     2.3     0      .0     0      .0     6     1.8
Period 1  170 2388        2816     4.0     0      .0     0      .0     6     3.5
Period 2  170 2388        2816     7.4     0      .0     0      .0     6     7.3
Period 3  170 2388        2816    10.8     0      .0     0      .0     6    11.7
Period 4  170 2388        2816    13.0     0      .0     0      .0     6    14.6
Period 6  170 2388        2816    15.6     0      .0     0      .0     6    18.6
Period 7  170 2388        2816    18.7     0      .0     0      .0     6    24.0
Period 8  170 2388        2816    22.4     0      .0     0      .0     6    31.9
Period 9  170 2388        2816    26.9     0      .0     0      .0    12    19.3
Period10  170 2388        2816    32.3     0      .0     0      .0    16    17.4

          --Non-Inter Rsp Time--  -----Non-Inter------  Release  Number of
Period     Local     LAN     WAN   CPU Util    Trans/Hr   Level   Processors
*current      .0      .3      .0        2.2       50227  V4R4M0      2
Period 1      .0      .3      .0        3.9      100274  V4R4M0      2
Period 2      .0      .3      .0        7.3      200330  V4R4M0      2
Period 3      .0      .3      .0       10.7      300397  V4R4M0      2
Period 4      .0      .3      .0       12.8      360477  V4R4M0      2
Period 6      .0      .3      .0       15.4      432572  V4R4M0      2
Period 7      .0      .3      .0       18.5      519087  V4R4M0      2
Period 8      .0      .4      .0       22.2      622904  V4R4M0      2
Period 9      .0      .4      .0       26.6      747485  V4R4M0      2
Period10      .0      .3      .0       31.9      896982  V4R4M0      2
```

Figure 107. Analysis summary in BEST/1

Note that the proxy caching function contributes a significant amount of CPU utilization. For period 7, as an example, there is just under 16% CPU utilization for the equivalent of about 140 object hits per second. Note the number of disk arms for periods 9 and 10. Our BEST/1 model indicates that we need to add disk arms and an IOP to keep resource utilization below recommended intervals.

You should also look at the storage pool data for main memory utilization, plus the communications IOP utilization. The final period indicates almost 900,000 transactions per hour and a CPU utilization of 32%. Looking at the Communications Resource report (Figure 108), you can see that the current communications IOP infrastructure should handle the load.

```
                        Display Comm Resources Report

Period:   Period10

                                          Overhead  Rsp Time per  Nbr of  Line Speed
Resource        Type          Util          Util     Trans (Sec)   Lines  (Kbit/sec)
CMB01           *MFIOP        15.8
  LIN02         *IPCS          5.1
    HTTPTEST    *LINE          6.1          .1           .76        1      16000.0
CMB03           *MFIOP        20.3
  LIN04         *IOA            .0
    TRNLINE     *LINE          5.9          .1           .08        1      16000.0
```

Figure 108. Communication resource report

In summary, we can conclude that the AS/400 HTTP server operating in a proxy cache environment will require some level of CPU, disk arm, main memory, and communications IOP resource. In all likelihood, we need to add this incremental workload to other HTTP server workloads that we expect on this particular machine.

As stated in Chapter 6, the purpose behind expending AS/400 server resources for the proxy caching function is to give end users a faster response time to Web page requests. In practice, you want to monitor the efficiency of your proxy cache server by comparing the proxy log results with the proxy caching log results to ensure that you are getting a return on your investment. You will find proxy caching most beneficial for a few heavily accessed Web sites whose content is moderately static. Additionally, you have to experiment in your own environment to determine if proxy caching, in fact, measurably reduces end user response time.

7.7 Server capacity planning tools

We have seen that thoroughly analyzing Web-based applications can be quite challenging because of the complexity and independent contributions from the client, server, and network resources involved.

On the AS/400 system, there are two means for collecting performance data: the Start Performance Monitor (STRPFRMON) system command and Collection Services within AS/400 Operations Navigator. This provides the basis for many performance management and capacity planning tools (IBM and third party).

7.7.1 IBM PM/400 and PM/400e

OS/400 includes the Performance Management/400 (PM/400) integrated function. PM/400 automatically activates the OS/400 Peformance Monitor and collects system utilization information, for example, CPU utilization, disk capacity and arm activity, response time, throughput, and application and user usage. That data is summarized and telecommunicated weekly to IBM via a customer dial-up communications line. IBM receives and retains the customer data for analysis.

Capacity planning and performance analysis reports and graphs are then returned periodically to the customer showing their AS/400 system's utilization and growth trends. PM/400 has been replaced by PM/400e.

PM/400e extends the PM/400 capability to provide improved management information about your system. It also allows the customer to view their respective reports and graphs on the Internet.

PM/400e gives you capacity planning and performance analysis reports and graphs that provide a crisp picture of your current system operating efficiencies. If you qualify, you can receive this service for free. Based on current trends, these reports let you know when to consider rectifying an approaching capacity planning problem. For more information or to register to use this service, see the Web site at: `http://www.as400.ibm.com/pm400/pmhome.htm`

7.7.2 IBM Performance Tools for OS/400

The IBM Performance Tools for AS/400 licensed program provides a set of reporting, analysis, and modeling functions to assist the user in managing the performance of the AS/400 system. It provides printed and online reports, either in graphic or tabular form, that portray the performance and utilization of AS/400 systems and applications. The Performance Advisor function assists the user in analyzing system performance, diagnosing performance problems, pinpointing performance bottlenecks, and providing recommendations to eliminate those bottlenecks.

The Performance Tools product requires a base package, and one of two optional packages, such as agent or manager. The agent provides a low cost tool for collecting, converting, displaying, and working with current and historical performance data. It also includes the Performance Advisor. The manager provides a full function analysis and planning solution. For example, performance system reports, component reports, transaction reports, and BEST/1 sizing and capacity planning analyses shown throughout this publication are included with the manager. The manager also gives you advanced tools, such as the Performance Explorer trace analysis tool.

7.7.3 Other solutions

There are a number of performance management and capacity planning products available for analyzing AS/400 data. For example, refer to the *AS/400 Magazine*'s Web site and look for the November, 1998 issue. It has over 20 references to IBM and other third-party products. Or, you can research the products on the search 400 Web site. These sites are at:

- `http://www.as400magazine.com`
- `http://www.search400.com`

For highly specialized or customized performance analysis, management, and capacity planning, you can write your own applications. When the AS/400 Performance Monitor runs, it records a wealth of data in standard AS/400 DB2 database files. This information is accessed by Performance Tools for OS/400, but can also be used for your own custom reports and queries. For more information on AS/400 performance data files, see *AS/400 Work Management manual*, SC41-5306, for your particular operating system release.

7.8 Other considerations

The capacity planning methodology has focused on gathering a relevant snapshot of real data obtained from our Web application environment for the client, server, and network environments. It also looks at correlating it with user-based requests, such as Web hits and transactions. This obviously represents the past, but we can learn from it in predicting the future for our particular environment. We have also indirectly dealt with the complex topic of forecasting future growth, but with few specifics outside of plugging in uplift factors into the BEST/1 capacity planning tool.

7.8.1 Growth

The most important factors in capacity planning deal with predicting future workloads and application environments for the Web site. Whether you plan on 5% growth per year or per hour has a major impact on the server and network infrastructure. It also affects the funds and other resources that you request from our organization's financial group. The resources that you request and what we get are often quite different.

Our intent is not to recommend complex mathematical forecasting techniques, such as exponentially weighted averages or modeling seasonal effects with sinusoidal or other goodness of fit tests, which often have minimal practical benefit. By the same token, blindly relying upon a growth rate of x% because that's what a favorite computer magazine says is also usually not valid. The most beneficial forecasting correlates non-random organization events (earnings reports, new products or services available, marketing campaigns or product reviews in the trade press, holiday shopping season) to changes in the Web site traffic. Let's look at an example of hits over a fairly lengthy period of time (Figure 109).

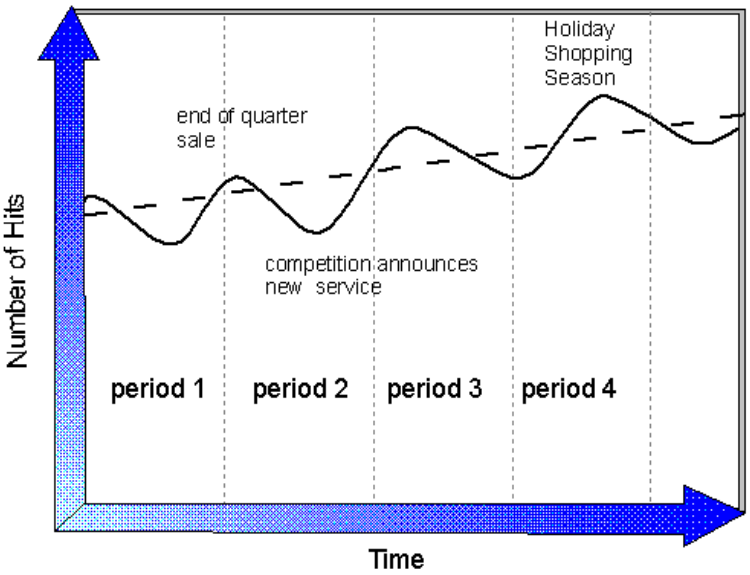


Figure 109. Measured Web site hits per day over time

You will generally see a great variation in hits per day. Tuesdays may be your most busy day, peak traffic tends to occur at 4:00 p.m. Eastern time, etc. Often

times peak and trough conditions occur because of predictable events, or lack thereof. Also, we have drawn a linear goodness of fit through the data for a rough idea of the growth rate.

From a simple analysis such as this, we can determine if the linear growth rate is accurate, or possibly too conservative because we intend to increase advertising and accelerate our Web-based application deployment.

Besides short term and long term growth, we also need to understand the peak load we need to plan for. As we have seen numerous times, looking at averages is easy, but there may be explicable and inexplicable situations where an overall average of 10 hits per second and 10% CPU utilization periodically reaches 20 hits per second and 30% CPU utilization.

7.8.2 User expectations

In a LAN environment, users are often accustomed to sub-second application response times. If deploying a Web-based transaction oriented application in this intranet environment, you need to be close to the existing application response time, or offer increased functionality and value add. Otherwise, enabling "client independence" or added color and graphics doesn't mean you are delivering value add to your organization and making people more productive.

The Internet environment has given rise to the joke that WWW stands for *world wide wait*, and sometimes it seems that way. We mentioned earlier that a good rule is that users give you eight seconds before deciding to stay, click through to another page on your site, or go to another site. You have to experiment to find that middle ground where your Web site and applications have enough graphics and other "sizzle" to make them appealing, but not so much that the wait becomes unbearable.

7.8.3 End-to-end response time

Chapter 6, “Sizing Web-based applications” on page 91, on sizing goes into great detail on response time measurements. We also mention that this is complicated by factoring in service and wait time for the server and network. What you may need to do is periodically (and objectively) sample the response time to your Web site and applications. You can do this simply by randomly accessing your site as a prospective customer or user would and recording the response time. You can purchase an application management suite that monitors response time and even sends an e-mail or pager message when certain thresholds are exceeded. For Internet sites, a number of third-party services will monitor response time to your Web site and provide detailed reports based on time of day, etc.

The important next step is to correlate these results with the Web site general health readings, such as CPU utilization, network utilization, or page fault rate on the AS/400 Web server. Eventually, you should be able to establish a set of operating guidelines or "leading indicators," such as those shown in Table 29 on page 166.

Table 29. Sample response time monitoring criteria

Scenario	Low response time	Average response time (30% CPU, 20% network utilization)	High response time (60% CPU, 40% network utilization)
Page 1	2 seconds	3 seconds	4 seconds
Page 2	4 seconds	7 seconds	9 seconds
Page 3	8 seconds	12 seconds	17 seconds

Use these expected response time criteria to establish control charts (Figure 110 on page 166) that help you understand the overall health of your Web site and applications.

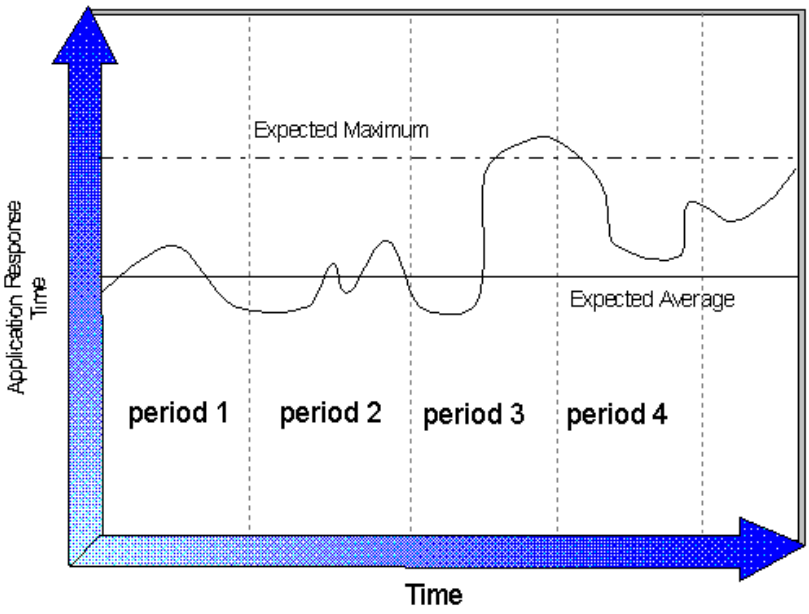


Figure 110. Sample application response time tracking

7.9 Clustering and load balancing

As your Web applications grow in popularity and maturity, you may need to periodically increase your investment in server infrastructure. A hallmark of the AS/400 server is its scalability. You can easily upgrade the CPU, main memory, disk arms, and communications resources. Eventually, you may have to reach a decision on whether to add another small or mid-sized machine, or move to one, much larger, machine. Each option has its pros and cons. We will not debate them here.

7.9.1 Load balancing solutions

If you deploy multiple servers, whether in a server farm environment or an N-tier application server environment, a load balancing solution may be needed. There are numerous techniques ranging from simple to complex and inexpensive to very expensive. We discuss some of the common families of solutions in this section.

7.9.1.1 Virtual IP address

The concept behind a virtual IP address is to provide load balancing and failover. The end users accessing your site will specify http://www.mycompany.com at their browser. This equates to a valid IP address, such as 205.206.207.208. You use virtual IP addresses on servers as a pseudo address. You may have two network connections to the Internet and a network card in our AS/400 system for each one. You assign a real IP address to each physical interface on the AS/400 system (such as 10.20.30.40) and bind this to the virtual IP address of 205.206.207.208, for example. This enables you to have a common domain or host name for which your servers can be aliased. Web page requests can come in to your site and be forwarded to any appropriate network interface on any of your servers. OS/400 has supported this feature since V4R3. See Figure 111 and Figure 112 for an example.

```
                    Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . > '205.206.207.208'
Line description . . . . . > *VIRTUALIP    Name, *LOOPBACK...
Subnet mask  . . . . . > '255.255.0.0'
Associated local interface . . .   *NONE
Type of service  . . . . . .   *NORMAL       *MINDELAY, *MAXTHRPUT...
Maximum transmission unit  . . .   4096         576-16388, *LIND
Autostart  . . . . . .   *YES          *YES, *NO
PVC logical channel identifier                001-FFF
             + for more values
X.25 idle circuit timeout  . . .   60           1-600
X.25 maximum virtual circuits  .   64           0-64
X.25 DDN interface . . . . . .   *NO           *YES, *NO
TRLAN bit sequencing . . . . . .   *MSB          *MSB, *LSB
```

Figure 111. Creating a Virtual IP address

```
                    Add TCP/IP Interface (ADDTCPIFC)

Type choices, press Enter.

Internet address . . . . . > '10.20.30.40'
Line description . . . . . .   httptest      Name, *LOOPBACK...
Subnet mask  . . . . . .   255.255.255.0
Associated local interface . . .   '205.206.207.208'
Type of service  . . . . . .   *NORMAL       *MINDELAY, *MAXTHRPUT...
Maximum transmission unit  . . .   4096         576-16388, *LIND
Autostart  . . . . . .   *YES          *YES, *NO
PVC logical channel identifier                001-FFF
             + for more values
X.25 idle circuit timeout  . . .   60           1-600
X.25 maximum virtual circuits  .   64           0-64
X.25 DDN interface . . . . . .   *NO           *YES, *NO
TRLAN bit sequencing . . . . . .   *MSB          *MSB, *LSB
```

Figure 112. Binding a physical IP address to the virtual IP address

Using a virtual IP address is advantageous in that it is requires a minimal, if any, investment in hardware, software, and services. It allows inbound traffic load balancing by letting routers assign different metrics to each actual interface. In other words, the router table contains entries, such as for address

205.206.207.208. The next hop is 10.20.30.40, with a routing metric of x. It does not do any querying of network or server resource utilizations or availability to optimally allocate workloads. Depending on your router configuration, all traffic can conceivably go to the same network card on the same server if it is misconfigured.

7.9.1.2 Round robin Domain Name Services (DNS)

Round robin DNS (Figure 113) address load balancing at the point where host names are translated into IP addresses. It involves rotating through a table of alternate IP addresses to assign loads to multiple servers.

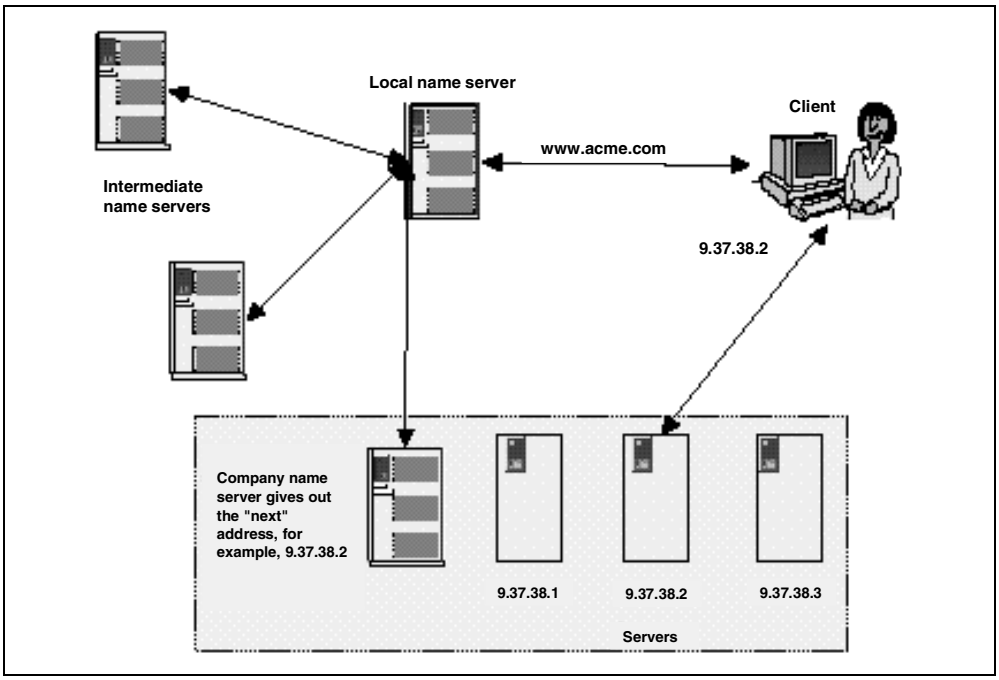


Figure 113. Example of a round robin DNS

This provides a degree of resource balancing and is transparent to the client. However, it requires a fair amount of network resource planning and has no awareness of server availability or workload.

7.9.1.3 Traffic shaping solutions

There are several approaches in place today, often vendor specific, for more intelligent load balancing. Some are software-based products that run on standard routers, or on high-speed workstations such as an RS/6000. An example is the IBM Network Dispatcher (Figure 114), which runs on a variety of server machines and routers, such as IBM 2210, 2212, and 2216 devices. The IBM Network Dispatcher is a software-based solution for load balancing requests among a group of HTTP, FTP, or other TCP-based applications. It is comprised of three components:

- **Executor**: Processes packets from clients to a defined virtual IP cluster address.
- **Advisors**: Monitor server availability and load.
- **Manager**: Uses information from the executor and advisors to assign relative weights to the servers.

From an implementation perspective, each server in the cluster either binds a physical address to the cluster address (1.1.1.1). Or it defines this cluster address as an IP loopback address to respond to packets destined for the cluster.

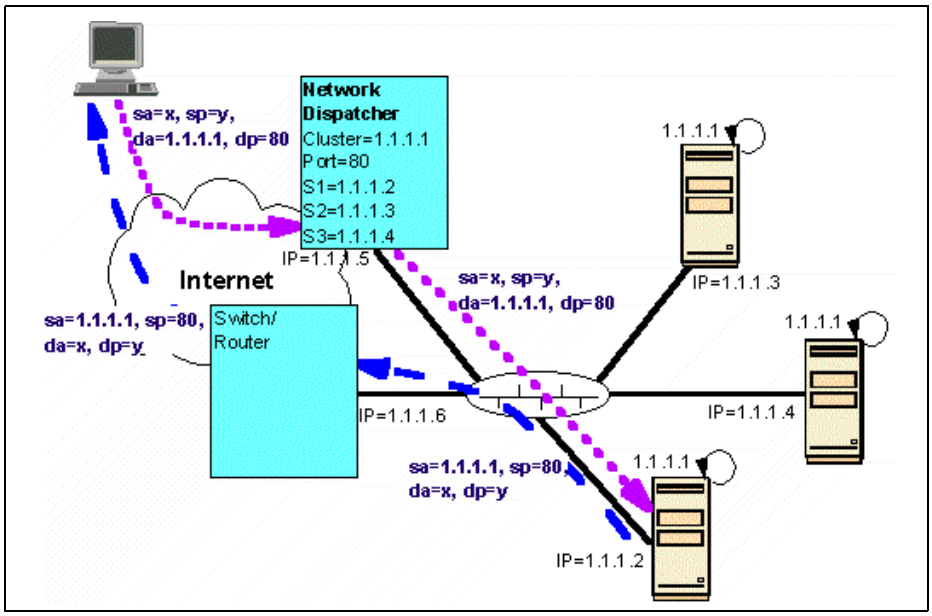


Figure 114. IBM Network Dispatcher example

For an overview of the IBM Network Dispatcher products, see the Web page at: `http://www.networking.ibm.com/white/serverload.html`

Another common approach involves LAN switches operating at the transport layer (OSI layer 4) or higher. These devices typically inspect the IP datagrams and make a route selection based on a predetermined criteria, such as "Send the next datagram for TCP port 80 to the listening host with the shortest PING response time." They can be used for HTTP and other application environments, such as SMTP. Plus, some can provide a rudimentary quality of service functions. These products and their management environment are typically quite vendor specific, which may be a consideration in your environment.

Still another, relatively new, approach involves traffic shaping appliances. Generally, these are application specific, for example, an appliance dedicated solely to HTTP traffic. These also tend to be vendor specific and typically deployed by Internet Service Providers or extremely large Internet or intranet sites.

The software-based solutions tend to be the least expensive and most flexible. The hardware-based solutions tend to be more expensive and much faster, operating at wire speeds. For a general overview of Internet traffic management, see the Web site at: `http://www.itmcenter.com`

7.9.2 N-tier architectures

The previous load balancing topics dealt primarily with solutions operating at the network level to route traffic to an appropriate server. You can also balance the

load across servers by using an application server architecture. These have important attributes, for example:

- The HTTP Web server infrastructure and workload can be managed separately from the transaction application infrastructure and workload.
- Application servers provide a wealth of connectors to enable extending transaction-oriented applications to a Web environment.
- Application servers can provide a pool of pre-started connection jobs, such as JDBC connectors, that can greatly improve overall performance.
- Application servers often have their own built-in load balancing capability. For example, the IBM WebSphere Application Server Performance Pack provides this and a number of performance management tools.

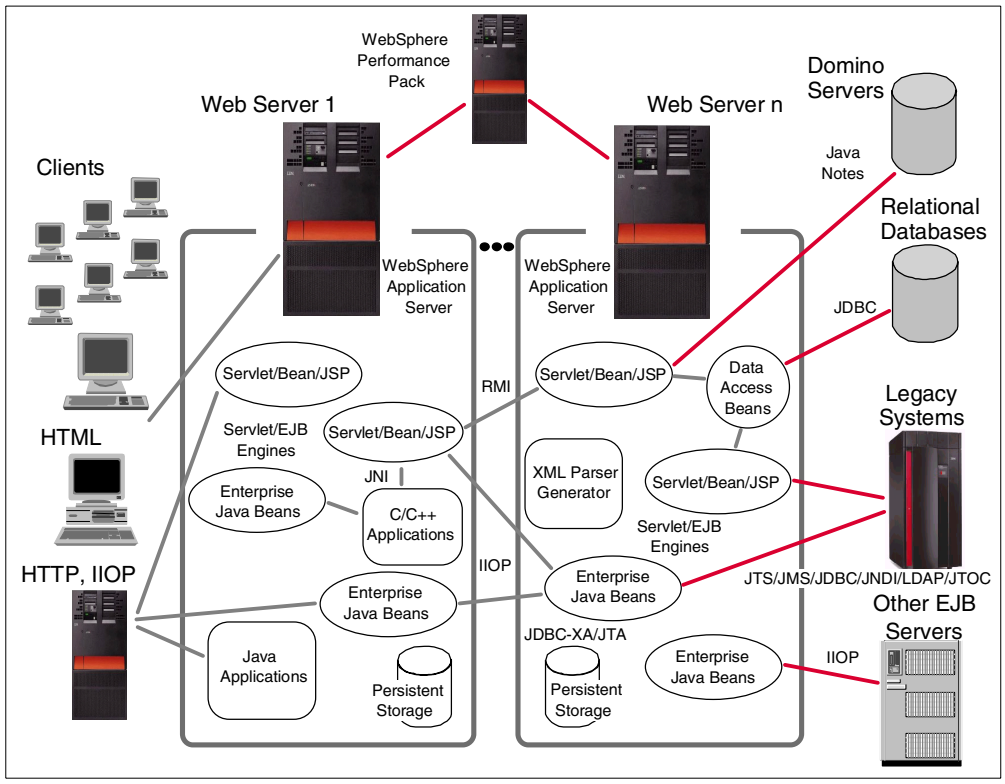


Figure 115. IBM WebSphere application server architecture

The application server concept provides a logical, software-based three-tier architecture that can physically be deployed in a two- or three-tier hardware environment. For example, the same AS/400 system can be the application server and database server, or these can be on separate machines. State of the art application servers, such as IBM WebSphere, include their own performance and load management capabilities, such as the WebSphere Performance Pack (Figure 116). This offers a wealth of tools for performance management and capacity planning, such as:

- Quality of service implementation via user classes and service classes
- Rules-based request routing for better peak load management
- Integrated IBM SecureWay Network Dispatcher for load balancing and high availability

- State of the art proxy caching
- Extensive logging and statistics that can be accessed from a command line or graphical user interface

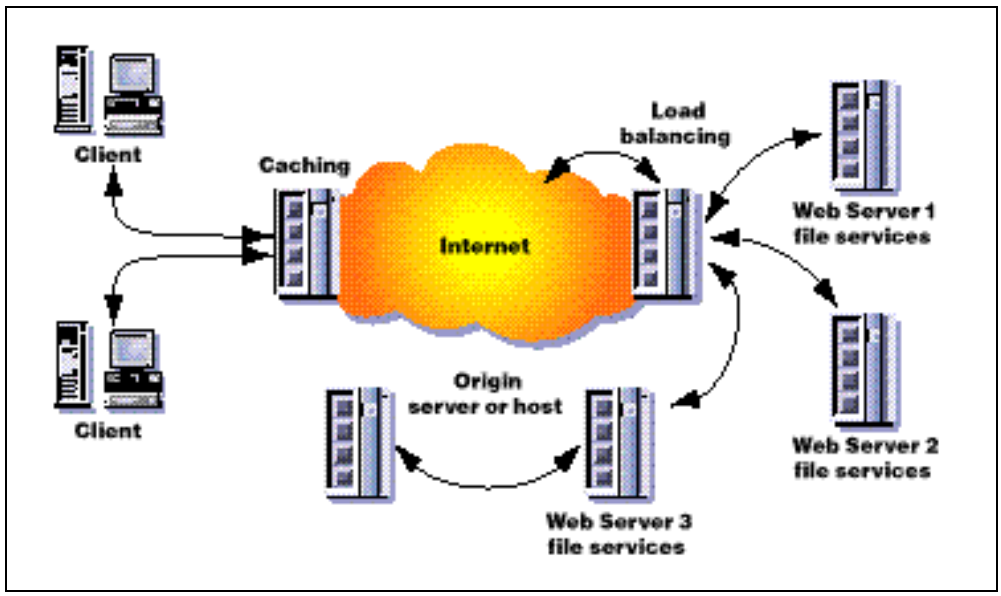


Figure 116. IBM WebSphere Performance Pack load balancing

Multiple-tier application performance considerations are complex, whether you have a physical two- or three-tier application server and database server solution. For physical three-tier architectures, the sizing and capacity planning steps will be especially difficult given that application processing will take place on multiple machines. There will be a small amount of overhead for the application management and connectors tasks. Plus, network traffic considerations between the servers will complicate matters greatly. The intent, of course, is that splitting the load among the servers gives a better response time than having all the logic on one server. As stated many times, you have to manage this for your particular environment. We highly recommended that you use the appropriate performance management package for your particular application server.

7.10 Summary

Capacity planning involves careful analysis of the server and network resource load and using this data and application transaction data to determine the necessary resources to meet a given service level. For your particular Web application environment, you need a thorough understanding of the actual transactions, such as static or dynamic pages, security factors, such as SSL, and the size and number of objects served. These can be obtained from the server access logs.

We also showed a number of examples using the AS/400 Performance Monitor data to understand key server resource usage (CPU, disk arms, main memory, and communications IOPs). The BEST/1 tool can be used to model any number of future scenarios for Web application endeavors.

Capacity planning for network resources is a complex subject in and of itself. You may need to consult with network specialists, especially if you choose to redesign or change the architecture of your communications infrastructure.

Each of these topics, individually, has a bearing on the overall key metric, response time. You may wish to pursue an application monitoring solution or employ a third-party monitoring service to accurately and objectively track this metric. Also, the multiple-tier application server architecture should be looked at closely if you plan a complex, transaction-oriented Web site.

Keep in mind that Web site data collection and analysis and capacity planning techniques must be an ongoing business process (Figure 117). They are not a once in a while event that is done grudgingly. Your Web application environment will be subjected to constant changes, uncertainty, and factors outside of your control. You do not want to log each and every Web page request to your site and may not have the AS/400 Performance Monitor running at all times, as your site matures. Rather, periodic sampling of these and other appropriate statistics is a better choice for striking the fine balance between too much and too little data.

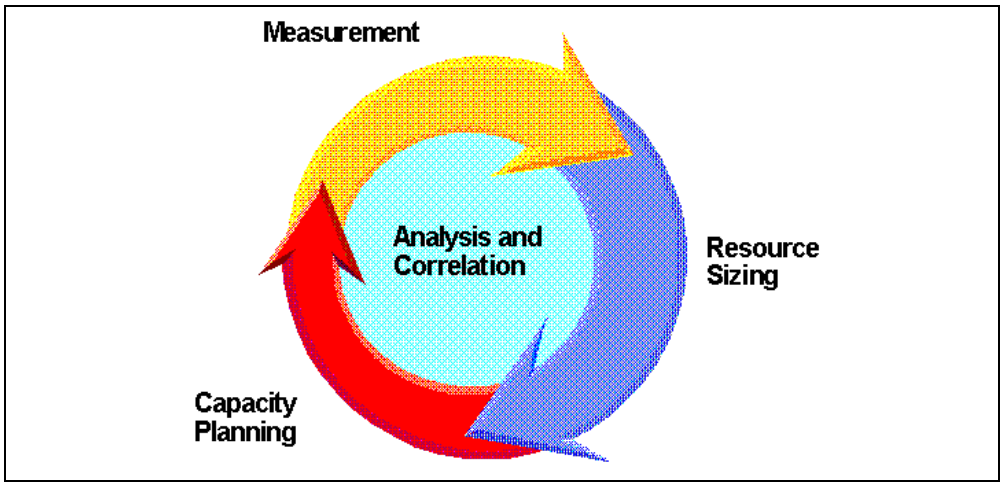


Figure 117. Measurement, analysis, correlation, sizing, and capacity planning

In other words, capacity planning is a journey rather than a destination. It also is one of several essential ingredients for having a successful Web site that performs at a price your management can accept.

Chapter 8. Web application performance considerations

This chapter deals with several popular Web application environments. Each application model or product is described in terms of general operation. Then, more specific details are provided regarding performance considerations.

8.1 Server Side Includes

This section covers a set of techniques called HTTP Server Side Includes (SSI). These techniques provide an easy to implement and easy to administer means of adding dynamic content to your HTML-based applications. These techniques do not require a heavy investment in server programming, or deal with multiple Web browser environments and varying degrees of support for scripts or Java applets. The IBM HTTP Server for AS/400 with V4R3 and later releases provides an easy-to-use, robust implementation of Server Side Includes.

Server Side Includes are a means to easily insert dynamic content from a Hypertext Transport Protocol (HTTP) server to a Web browser. The content can be defined in a static HTML page, or via a dynamic HTML creation process, such as a Common Gateway Interface (CGI) program. For example, you may want to have date and time information from the server displayed on your Web pages. You may have Web page components (HTML text, graphics, page headers and footers, etc.) that are deployed in multiple instances on your site. You may also want to minimize the impact of making changes without purchasing complicated, proprietary content management products.

8.1.1 SSI example

In the following example, we use several SSI tags to create dynamic HTML page content, served from the AS/400 system. For a more complete discussion of Server Side Includes and a list of functions supported on the AS/400 HTTP server, see the *Web Programming Guide V4R4*, GC41-5435.

```
<html><title>AS/400 Server Side Includes Example #1</title>
<!--      insert the page header defined in file headpage.html   -->
<!--#include virtual="/home/headpage.html" -->.  It is now
<!--      insert the server's date and time   -->
<!--#config timefmt="%n %A %D %T"  -->
<!--#echo var=DATE_LOCAL --> in Rochester.
<body>
<!--      insert the today's specials   -->
<!--#include virtual="/home/today.html" -->
<!--     display the size and date modified for  some of the resources on our site  -->
<p>File ic400.gif: <!--#fsize file=ic400.gif  -->,  <!--#flastmod file=ic400.gif  -->
<br>Document home.html: <!--#fsize file=home.html  -->,  <!--#flastmod file=home.html  -->
<!--  Let's set  variables, then specially format text containing them    --><p>
<!--#set var="author1" value="George Weaver" -->
<!--#set var="author2" value="Saeid Sakhitab" -->
<!--#set var="credits" value="Who wrote this\?\tIt
was<br>&author1;<br>&author2;" -->
<!--#echo var="credits" -->
</html>
```

Figure 118 on page 174 shows an example of the finished product.

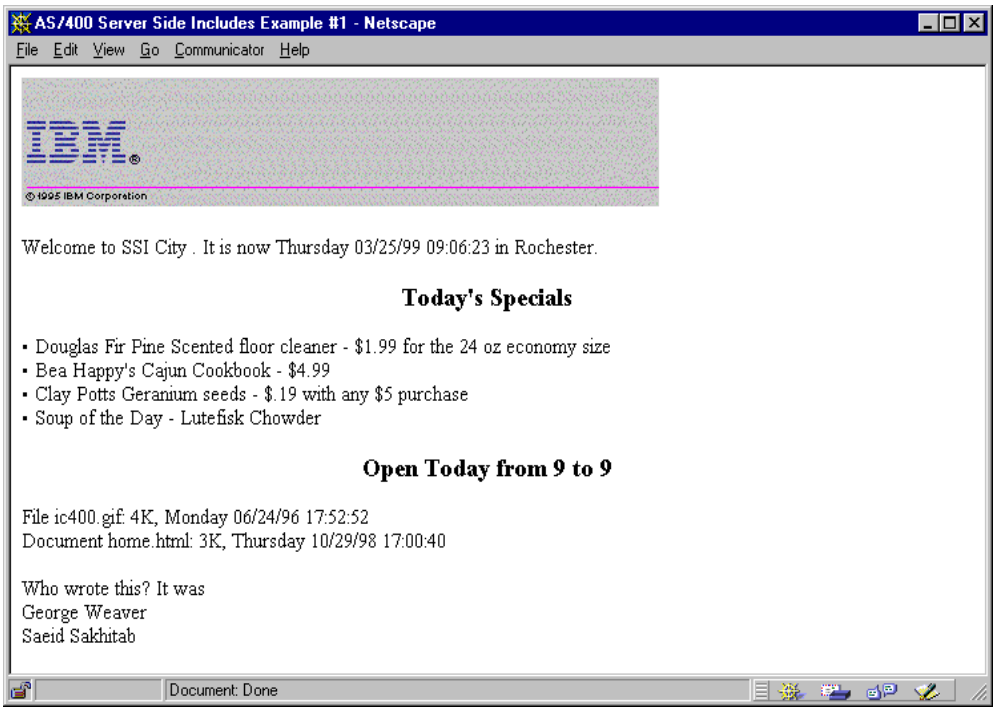


Figure 118. SSI output example

8.1.2 Server Side Includes implementation

Notice that the SSI tags are essentially a special form of an HTML comment. HTTP servers must be specifically enabled to interpret these directives and deliver the appropriate results. The server basically analyzes the page for includes, processes the includes, and delivers the page to the user's browser with the rest of the HTML document. If the server is not SSI-enabled, the SSI tags are treated as comments. There is a trade-off, of course. Generating dynamic content is very flexible. However, there is additional server processing taking place. Any HTML page can have Server Side Sncludes. We ideally only want to incur the processing overhead for HTML pages that actually have them.

Another implementation aspect is that SSIs are not limited to HTML files. A CGI program or Net.Data query can also contain SSI directives in the output datastream. There is also a specific Multipurpose Internet Mail Extensions (MIME) type for server side includes. Here is an example AddType directive you would define in an HTTP server configuration file:

```
AddType .shtml text/x-ssi-html 8 bit 1.0
```

If your CGI program generates HTML output, you must preface the SSI portion of the output datastream with an appropriate MIME type statement that defines the text/x-ssi-html MIME type. Otherwise, your SSI directive will be treated as an HTML comment.

8.1.3 SSI performance implications

Server Side Includes involve a server process that looks for SSI tags in the HTML and then embeds the appropriate data in the output sent to the browser. This requires a certain amount of server workload that you want to manage properly. It

makes sense to incur these workload tasks only on pages that have SSI tags. In practice, you generally want to identify SSI-enabled pages separately from other HTML pages by assigning a different file extension (usually .shtml or .htmls). When you enable Server Side Includes for specific file extensions, the tag search process happens for *every* document of this extension type. Obviously, you do not want all your HTML pages to incur this extra processing overhead unless *every* page has Server Side Includes.

8.1.4 AS/400 setup for SSI

To enable the AS/400 HTTP server to support SSIs, you must edit your configuration file. A new configuration file directive, imbeds, has been added to provide this support. You have the option of specifying for each HTTP server instance whether SSIs should be enabled (the default is not), static pages and CGI programs are to be SSI-enabled, and standard HTML pages are to be SSI-enabled.

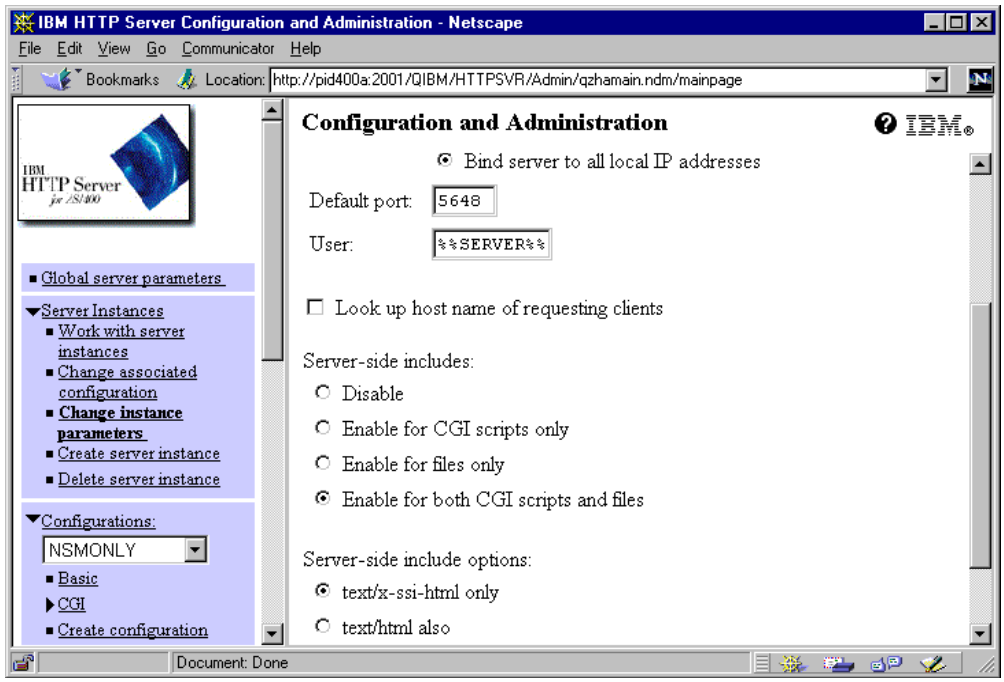


Figure 119. SSI setup on the HTTP server

In the example in Figure 119, we selected Server Side Includes options for text/x-SSI only. This means that only files with the MIME type indicated will be analyzed for SSIs. Standard HTML files will not be analyzed in this example. Generally, you would use the file extension .shtml or .htmls to indicate SSI-enabled content, which would tell the HTTP server to incur the SSI processing required.

In our implementation example, the following two lines were added to the configuration file to enable SSI support for .htmls and .shtml files:

```
AddType .shtml text/x-ssi-html 8 bit 1.0
AddType .htmls text/x-ssi-html 8 bit 1.0
```

We have enabled Server Side Includes for static pages and CGI programs, but not for standard .html or .htm files. This will also appear in the HTTP configuration file:

```
Imbeds    On    SSIOnly
```

To recap, we have configured the AS/400 HTTP server to process Server Side Includes tags for CGI programs and static pages. However, only files with extensions .htmls or .shtml will be analyzed and incur the processing overhead

8.1.5 SSI pages and caching considerations

We discussed earlier how caching generally improves Web application performance. We said that it eliminates the need to download objects that have not changed since the last retrieval. Conceptually, the browser or proxy compares the file modification date on the server to the date in the cache and refreshes the content if necessary. Consider the sample Web page in Figure 120.

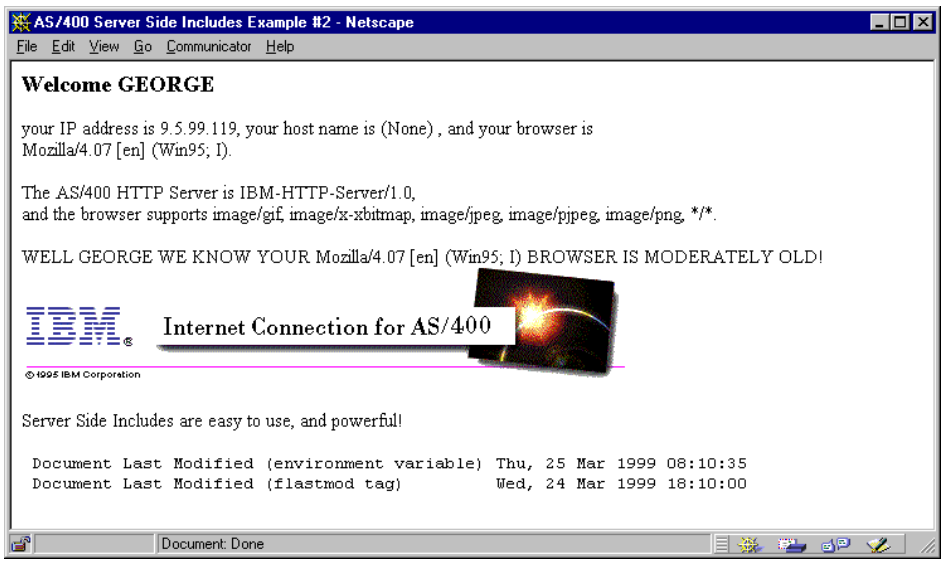


Figure 120. SSI example

The SSI-enabled page includes two SSI tags to put the current document's modification date at the bottom of the page. The SSI tag `#echo var="LAST_MODIFIED"` considers the date the page is created as current, since the document is dynamically constructed. The SSI tag `#flastmod file=ssiex2.shtml` refers to the edit date from the AS/400 IFS for the object ssiex2.shtml. Which is correct? Most people would probably say the latter. Again, many Web sites require the file modification date to be displayed on every page. Server Side Includes offer an easy means to do that. However, to the cache on your Web browser or any proxy server cache, the modification date is the former. This may not make sense initially. Since SSI-enabled pages are dynamically constructed, your perspective may be that the page's effective modification date and time is the moment it was served. This is what the HTTP environment variable views as the last modified date, versus intuition telling us that it is the date the file was last edited. Certainly, this can also affect overall performance for the end user.

8.1.6 SSI recommendations

Consider these recommendations when using SSI:

- Understand the advantages and disadvantages of using Server Side Includes.
- The performance impact to the server for any given page is relatively small, but it adds up.
- Do not enable Server Side Includes for text or HTML document types.
- Limit Server Side Includes to special file extensions with a MIME type of text/x-SSI-html.
- Realize that Server Side Includes do not reduce the number of objects being served.

8.1.7 General Net.Data performance tips

Most Net.Data performance characteristics pertain to SQL, which is covered in the following section. Here are a few guidelines specifically for the Net.Data macro environment:

- Minimize the number of rows to be returned to the browser.

  Net.Data provides two functions, RPT_MAX_ROWS and START_ROW_NUM, to enable you to send a subset of rows in an HTML table back to the browser client. You would then add the Next and Previous buttons or tags to view other rows of the table.

- Avoid having too many built-in function calls within a row-block.

  You may be able to use a built-in function for an entire table, rather than invoking calls for each row or each cell in the table.

- Avoid using VLIST and NLIST, if possible.

  Referencing these variables requires extra processing to construct them from the report block table. It is faster to access individual fields such as V1 and N1.

- Use external, rather than in-line, Rexx programs.

  There is a slight performance penalty for having your Rexx programs within the macro, instead of in a separate file.

- Use the direct call environment.

  A recent set of PTFs (which includes V3R2, V3R7, and subsequent releases) provides a new language environment, Direct Call (DTW_DIRECTCALL), which enables calling AS/400 programs (for example, C, RPG, CL, Cobol, etc.). It also offers the ability to pass parameters to, and receive values from, the program that is being called.

  The major benefits of this support are the easy integration of existing programs with Net.Data and the performance improvement over the system language environment.

  The Direct Call feature of Net.Data makes it easy to call other AS/400 compiled programs from Net.Data. Prior to this enhancement, you needed to call a stored procedure to pass parameters and receive parameters. Or, you had to use the system language environment to call a program and pass values in environment variables (using the AS/400 system APIs QtmhGetEnv and QtmhPutEnv).

8.1.8 SQL tuning tips

Since Net.Data relies quite heavily on SQL, you need to be as efficient with it as possible. For example, stored procedures perform much better than dynamic SQL. Also, you get better performance from a database stored in single byte, versus double byte or Unicode. Refer to the following resources for more in-depth information:

- Chapter 22 in *DB2 for AS/400 SQL Programming Version 4*, SC41-5611, which gives tips and techniques on query performance, and information on the DB2/400 Query Optimizer
- The DB2/400 technical tips and techniques Web page for a variety of resources for optimizing your database applications:
  `http://www.as400.ibm.com/db2/db2tch_m.htm`
- The AS/400 Partners In Development site for a number of general SQL and database access tips and techniques:
  `http://www.as400.ibm.com/developer/client/performance/csperdg5.html`

8.2 Java servlet applications

Over the past several years, there has been an enormous increase in the interest and usage in Java. Java is an object-oriented programming language that is relatively easy to use. Application developers write the code and compile it into bytecodes that can conceivably run on any platform. Java is also a runtime environment which uses a Java Virtual Machine (JVM) to execute these byte codes.

You may already be familiar with Java applets, which are downloaded from an HTTP server and execute in the JVM of your Web browser. Java can also be used to develop server applications. This section discusses Java servlets.

8.2.1 Java servlet overview

The Java language and run time define a servlet application programming interface (API). Developers write server-based Java applications to this API, typically for deployment on an HTTP server. Servlets (Figure 121) are similar to applets. However, they run on a server and have no graphical user interface.

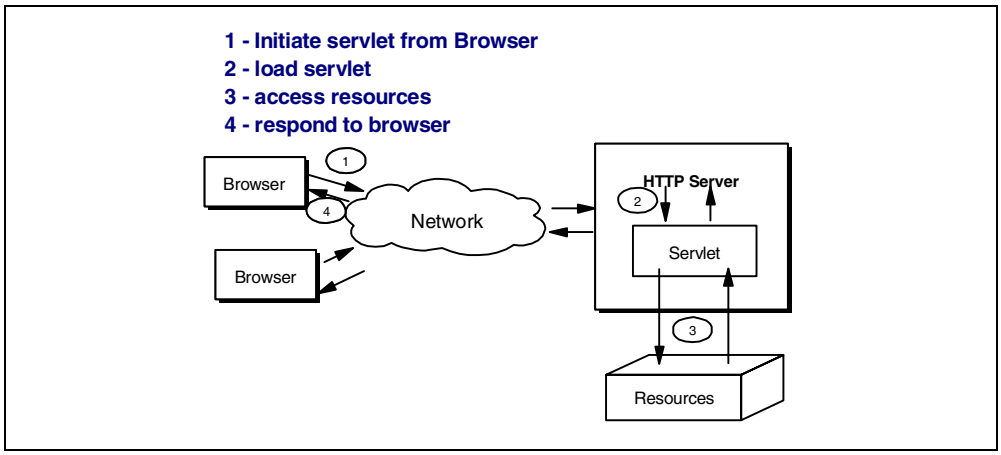


Figure 121. Servlet overview

Java servlet support is often wrapped under products commonly called *Web application servers*. WebSphere is an IBM strategic application server product and is supported on AS/400 releases V4R3 and later. The WebSphere application server (Figure 122) enables Java servlets in conjunction with the AS/400 HTTP server.

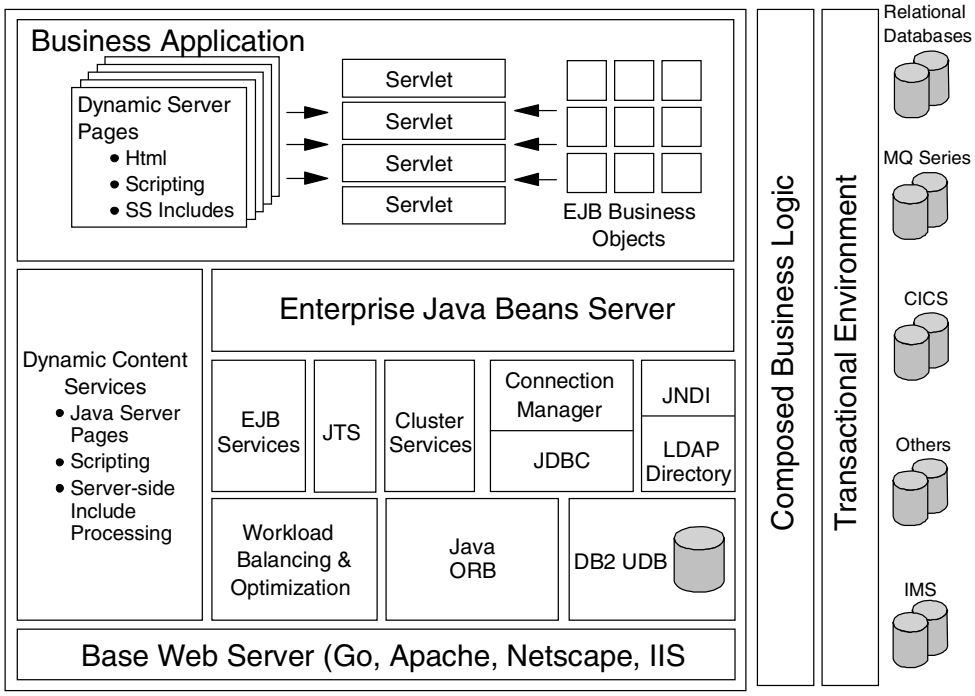


Figure 122. WebSphere application server architecture

Servlets apply in many Web application environments. They can be used similarly to CGI programs, in that they can process an HTML form from a user's browser, and return results, such as an account inquiry application. Today's application servers, such as WebSphere, also provide session management functions to facilitate transaction-oriented applications over the Web. Servlets, by definition, have state management built in and are an excellent way to extend new categories of applications to the Web or an intranet.

An in-depth discussion of developing Java servlets is beyond the scope of this redbook. Instead, our focus is on components that affect overall performance, primarily the server in this case. We discuss servlets from two vantage points: state aware and non-state aware.

8.2.2 Servlets non-state aware model

This non-persistent servlet model is analogous to the CGI environment we discussed earlier. It generally involves a Web browser-based application comprised of an HTML form. The user enters information to the form, submits the form, and expects a response. After the response is received, the transaction is considered finished. Examples include an account or order inquiry application, registering at a Web site to receive information, etc.

8.2.2.1 Browser or client considerations

In this non-persistent environment, the client generally receives a standard HTML document that is downloaded from the HTTP server. This document typically uses one or more HTML form elements to enable user input. In fact, the POST action looks much the same as that for CGI programs, for example:

```
<FORM NAME="MySearchForm" ACTION="/servlet/mysearch" METHOD="POST">
```

From a performance perspective, the client and network considerations are essentially the same as any static HTML page serving. The HTTP document and any images are all separate objects needing to be served, just as we described earlier. The server environment is not so straightforward, since we have to build the Web page, in addition to serving it.

8.2.2.2 Server considerations

In the typical Java servlet environment, the following process occurs. Our focus is on steps 4 and 5.

1. The user requests a standard Web page that invokes a servlet.
2. The HTTP server sends the Web page to the user.
3. The user enters information, such as their account number or search criteria, and submits an HTML form.
4. The HTTP server receives the form and passes the input parameters to the application server.
5. The application server processes the user's request and builds the appropriate HTML document to be returned to the browser.
6. The HTTP server sends the servlet-built document to the user's browser.

CGI has been the method used for creating dynamic Web pages. However, it has a number of limitations, which we discussed earlier. Servlets are a more efficient means for enabling dynamic Web applications. First, servlets are run under the auspices of the HTTP server process and do not require the overhead of additional job creation. On the AS/400 WebSphere implementation, this runs in conjunction with your HTTP server instance, in the QHTTPSVR subsystem. The servlet process also has direct access to system services, such as security. Servlets also remain resident after being initially loaded. They can be reused, unlike CGI programs. This helps provide a 4 to 10 times better throughput compared to CGI. It also minimizes the impact of step 4 in the above list. Most Web application servers are highly optimized to minimize the time needed to connect to the actual application logic.

8.2.2.3 Enabling servlet support in your server instance

The following example shows a set of directives in an AS/400 HTTP server configuration that is servlet enabled:

```
Service /servlet/* /QSYS.LIB/QHTTPSVR.LIB/QZHJSVLT.SRVPGM:AdapterService
ServerInit /QSYS.LIB/QHTTPSVR.LIB/QZHJSVLT.SRVPGM:AdapterInit
/QIBM/UserData/IBMWebAS/george/properties/server/servlet/servletservice/jvm.pr
operties
ServerTerm /QSYS.LIB/QHTTPSVR.LIB/QZHJSVLT.SRVPGM:AdapterExit
```

Our standard Web pages can invoke servlets with a POST HTML tag. We can invoke a servlet directly using a Web page address such as this:

```
http://myserver/servlet/myservletpage
```

By specifying the /servlet directory, it reverts to the actual directory defined in the Java classpath statement in your jvm.properties file, rather than the map or pass statements in your HTTP configuration file.

8.2.2.4 Servlet application considerations

Servlet applications can range from simply calling out an existing HTML page on the local server, through complex applications that read or write data to the AS/400 database or other objects. An in-depth treatment of these topics is beyond the scope of this redbook. However, there are a number of general guidelines to follow, plus many excellent references.

Java considerations

Certainly, there are a number of good references available on Java, in general, and Java deployment on the AS/400 server:

- JVM: Just in time (JIT) and static compilers, garbage collection
- Java application design: Objects and variables, jar and cab files, other tips

Application server considerations

Application servers are commonly deployed as the middle tier in a three-tier environment. In other words, they act as a liaison between the browser-based client and the data repository. The application server and data repository may be physically loaded on the same machine or on separate servers.

Database considerations

A high percentage of e-business oriented applications require the ability to create, replace, update, or delete persistent data in a relational database. Java applications, such as servlets, can connect to the AS/400 DB2 database in one of several ways:

- AS/400 Toolbox for Java JDBC driver
- AS/400 Toolbox Record Level File Access driver
- Native CLI JDBC driver

Many application servers, such as WebSphere, support pooled connections. These can be pre-started jobs, or started dynamically and made available for reuse. Creating a connection from scratch requires system resources. However, idle connections represent wasted resources. In practice, you may need to experiment and dynamically adjust this.

Another trade-off is Structured Query Language (SQL) versus native record access. SQL is portable and relatively easy to use. However, it is not as efficient as the Call Level Interface. If using SQL, refer to 8.1.8, “SQL tuning tips” on page 178, for more information.

8.2.3 Servlets persistent model

Servlets can also be used in a transaction-oriented environment. A good example is a shopping-enabled commerce site. You can use servlets to keep track of the goods in a virtual shopping cart.

8.2.3.1 Browser and client considerations

In this application environment, you typically use HTTP *cookies* to maintain state. A cookie is a local text file that contains information about your browser session at a particular Web site. Figure 123 on page 182 shows an example.

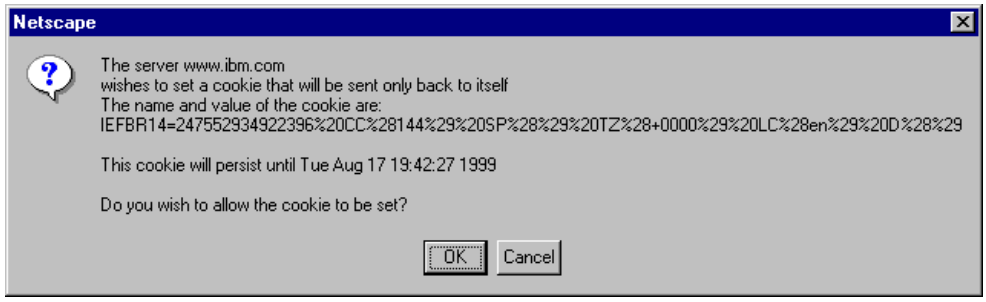


Figure 123. IBM Web site cookie setting

A server application reads the cookie and updates it as is appropriate. Typically, cookies must be enabled on the browser or persistence cannot take place. The performance impact on the client is negligible.

8.2.3.2 Server considerations

In addition to the topics discussed in 8.2.2.2, “Server considerations” on page 180, the server application is also responsible for some transaction management tasks. The application must be prepared to accept multiple service requests, such as a user purchasing goods online. Also, session state management is necessary for the transactions and typically is part of the application server.

8.2.3.3 Additional database considerations

In traditional client/server application transaction-oriented environments, the client is generally in control of the actual commit point, even on a remote data source. Since the nature of HTTP and a Web browser is a stateless environment, move these control tasks to the server, on behalf of the client.

Database servers, by nature, have a TCP/IP timeout period (typically two hours) which facilitates the stateless Web application model. However, contention can potentially occur because a browser application may have a lock on a particular resource for an extended period of time. There are three common techniques for dealing with database locking.

- **Optimistic locking**: This involves an incremental authority check at each stage of the transaction. However, updates are not actually made until the commit is requested.
- **Logical locking**: This involves an incremental authority check and an update flag set for the individual records. The permanent update is made at commit request time.
- **Incremental updates**: This involves making permanent updates as each phase of the business logic progresses. However, the application must be able to back out of the transaction if necessary (for example, a shopper decides not to buy the item).

Choosing a database locking mechanism involves trade-offs between the application and the business rules. For performance, optimistic locking and logical locking have the advantage of committing the entire transaction in one final step, versus the incremental update option.

In the transaction-oriented environment, having a pool of pre-started database connections is advantageous. You may have to do some experimentation with

this parameter. In one large benchmark done in Rochester, the number of connections tested ranged from 5 to 50 and the optimum appeared to be about 15 connections.

In some application servers, such as IBM WebSphere, pooled connections require a prepared or callable statement for invoking the connection. For heavy usage environments, you may want to write your own application that creates a pool of callable statements, versus connection objects. Similarly, the less authority checking you need to do, the better, since this uses less server and network resource and also extends the overall response time at the client.

8.3 Java server page applications

Java Server Pages (JSP) provide Web application developers with a means to easily separate the presentation and programming logic components of the application. Most commercial Web sites need their pages developed by teams of graphic designers and application programmers. JSP enables the Web page designers to focus on the user interface, site navigation, and usability of the site, without requiring an extensive programming background. Application developers can focus on the business rules and data integrity, without having to be artists. An example of the JSD transaction flow is shown in Figure 124.

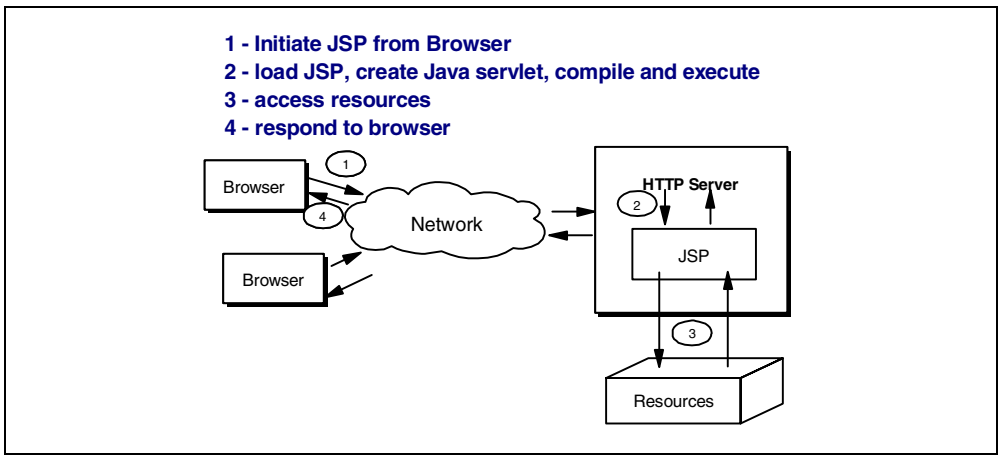


Figure 124. Java Server Page transaction flow

JSPs are similar to HTML documents with a JSP file extension. They typically contain HTML, Server Side Includes tags, Java programming code, and some JSP specific tags. As indicated in Figure 124, the Java code is actually run on the server as a servlet. The basic HTML is sent to the browser, along with the output of the servlet.

JSPs and Java servlets can be used separately or together. A fairly common implementation strategy uses servlets for transaction-oriented tasks, such as database reads or writes. The servlet can place its results in HTML for the end user, or in a Java bean component. A JSP can retrieve information from this "transaction" and then provide the appropriate user interface to the browser.

8.3.1 Performance implications

Since JSPs are essentially dynamically-created servlets, refer to the topics covered in 8.2.3.2, “Server considerations” on page 182. Because the Java code contained in the JSP must be compiled, then executed, the more code you include, the longer the response time is to the user and greater server load.

JSPs can run in a dynamic, interpreted mode which requires no configuration on the server. However, better performance can be achieved using the CHGJVAPGM command for all servlets created by the JSPs and specifying an optimization level of 40, for example.

If using Java beans in the JSP, the Beans.instantiate() method looks first for a serialized instance (.ser file). If this is not found, a new newInstance() function is called, which also impacts performance.

8.4 Net.Commerce applications

IBM Net.Commerce enables businesses to quickly, easily, and securely conduct electronic commerce on the World Wide Web. It helps companies integrate existing business process and legacy systems with the Web, and grow new Web-based businesses. Net.Commerce is scalable, open to customization and integration, and can handle any level of transaction volume. It comes complete with catalog templates, setup wizards, and advanced catalog tools to help you easily build effective and attractive electronic commerce sites.

8.4.1 Net.Commerce performance implications

Since Net.Commerce relies heavily upon Net.Data, many of the topics covered in 8.1.7, “General Net.Data performance tips” on page 177, apply to this discussion. In fact, the capacity metric for catalog browsing is almost identical to Net.Data HTML transactions.

Portions of Net.Commerce, such as the IBM Payment Server for ensuring secure credit card transactions over the Internet, can vary greatly due to network traffic and server capacity at the acquirer’s Web site. Similarly, using SSL and other security means must be carefully considered and factored in as appropriate.

8.4.2 Performance recommendations

Consider these recommendations for improving performance:

- Adjust the Maximum Active Value in the memory pool.

  Experiment with increasing the base storage pool activity level (QBASACTLVL) system value. Threaded tasks, such as the HTTP server and Net.Commerce, often have active threads even if they are not actually doing productive work. These compete for *BASE memory pool resources, the same as threads doing productive work.

- Adjust the Net.Commerce job priority.

  You can use the change class (CHGCLS) command for the QNETCOMM class in QNETCOMM library. The default priority is 25. This may affect other jobs on the system, so you may need to do some experimentation.

- Load the Net.Commerce tables to main memory.

  You can move AS/400 objects to main memory by using the set object access (SETOBJACC) command. For instance, Net.Commerce tables, such as products or services, could be loaded directly to main memory at strategic times, such as at the beginning of the day or week. There is a sample program illustrating this technique in *Net.Commerce V3.2 for AS/400: A Case Study for Doing Business in the New Millennium*, SG24-5198.

  Note: You must have ample main memory available before running the SETOBJACC command. Additionally, loading objects to main memory does not always improve performance.

8.5 Other applications servers

We have covered a number of specific e-business, Web-based application environments. Some of the first AS/400 Web application server implementations were for delivering 5250 terminal emulation to the browser environment. The earliest versions involved a 5250 datastream to HTML conversion. The AS/400 5250 Workstation Gateway component of the TCP/IP Connectivity Utilities program is an example. Later versions involved a Java applet running within the browser. The IBM Secureways Host on Demand product is an example.

8.5.1 Application characteristics

Since the objective of this class of products is to deliver a 5250 session to the browser user, there is a certain amount of interactive workload from a Telnet or Virtual Terminal API session.

If you are deploying a pure HTML-based solution, generally substantial server processing is necessary to convert the 5250 datastream to HTML, and vice versa. You may also, of course, incur server processing resources for sending the HTML pages and receiving requests. Typically, these requests are a CGI type application that tries to maintain the server state with a complex variable string or cookies.

If you are deploying a Java-based solution without HTML, you have a significantly lower server load. Typically, these applications require the end user to access a specific Web page and select an appropriate URL to invoke the applet. The browser's Java Virtual Machine has to be loaded at the client. From a Web server perspective, the HTTP server will deliver the appropriate .class, .jar, .cab type files containing the bytecodes to the browser. After delivering the program objects to the browser, typically the HTTP server's task is finished. The Java-based emulator typically uses a sockets-based application to interact with a middle tier server task that helps maintain the state of the 5250 application.

8.5.2 Performance implications

Certainly, the interactive workload must be figured in when doing any capacity sizings or planning. This is regardless of whether the solution is HTML or Java based. You can size this component as a normal Telnet interactive workload. This will substantially impact whether you choose an AS/400 system or server model for deployment.

For Java-based emulators, you need to size the amount of data to be downloaded to the browser client, and the frequency of the downloads. This may range from a very thin client to a very thick client and one or more megabytes.

Dynamic HTML-based emulator products can be a substantial load on the server. For the AS/400 Workstation Gateway, the server load is three to five times that of a straight 5250 Telnet session. These solutions also tend to require more communications resources (AS/400 IOP and network access).

8.5.3 Performance recommendations

Consider these performance recommendations for other application servers:

- Use 5250 to HTML conversion-based products sparingly, if at all.
- Avoid SSL and encryption on HTML conversion-based applications, if possible, since this will further magnify the server load and affect response time.
- Package your Java bytecodes appropriately, especially in .jar or .cab format. Avoid packaging rarely used classes in frequently downloaded files.
- Consider Host Access Class Libraries or Java beans to provide 5250 application functionality components to your browser-based applications.

Appendix A. IBM application framework for e-business

The e-business architecture model is based on an n-tier distributed environment. Any number of tiers of application logic and business services are separated into components and connected via industry-standard protocols, servers, and software connectors. The model identifies key elements for developing and deploying e-business applications. Each element is based on open, vendor-neutral standards, allowing you to substitute components from any vendor that supports those standards.

Figure 125 shows the relation of application servers to clients and data centers.

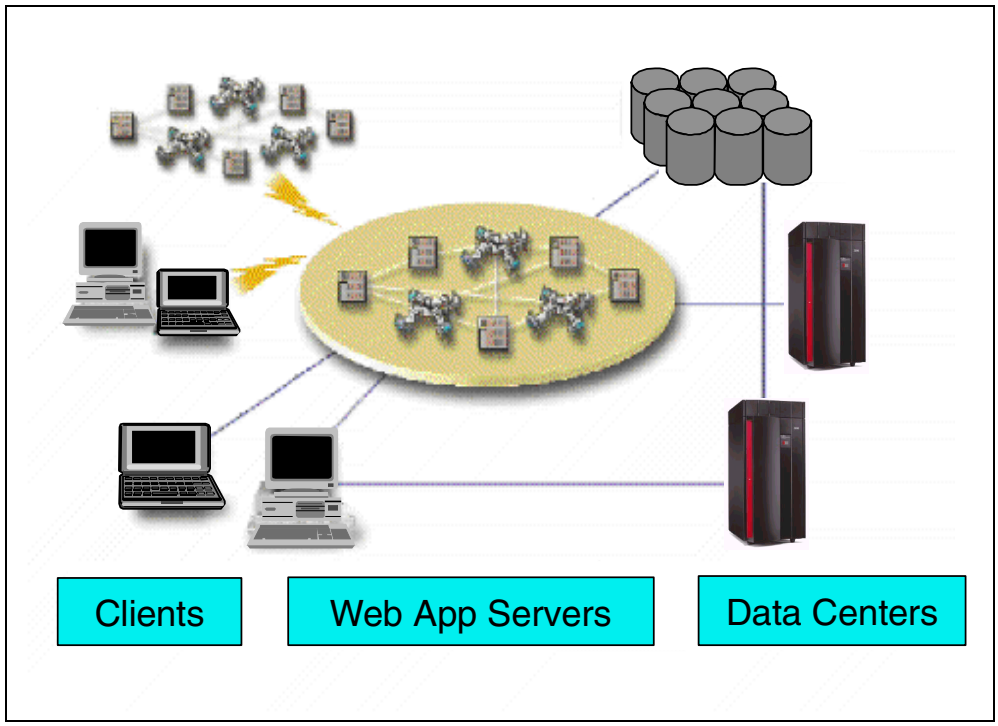


Figure 125. Clients to Services

The AS/400 system is a key component in the Web application server, as well as data center tiers. The Web application server is separate from, but depends on and is tightly integrated with, the HTTP server. In AS/400 terms, this is the IBM HTTP Server for AS/400 (5769-DG1), plus the Websphere application server (5769-AS1). The Web application server receives browser requests from the HTTP server, provides Java servlet or Java server page applications, manages state, and acts as a gateway to the backend data and applications.

The AS/400 system has supported the Websphere application server environment since V4R3. From a performance perspective, the AS/400 Performance Capabilities Reference lists the relevant metrics for V4R4 as follows:

```
.4 hits/second/CPW (not secure), .28 hits/second/CPW
```

These metrics reflect simple HTML page serving. More demanding applications involve database access, calling other applications, and possibly transaction

management. Estimating the appropriate hits/second/CPW requires using the techniques described earlier in this redbook to esimate the appropriate metric for your application scenario.

Figure 126 shows the functions of the middle tier, application system.

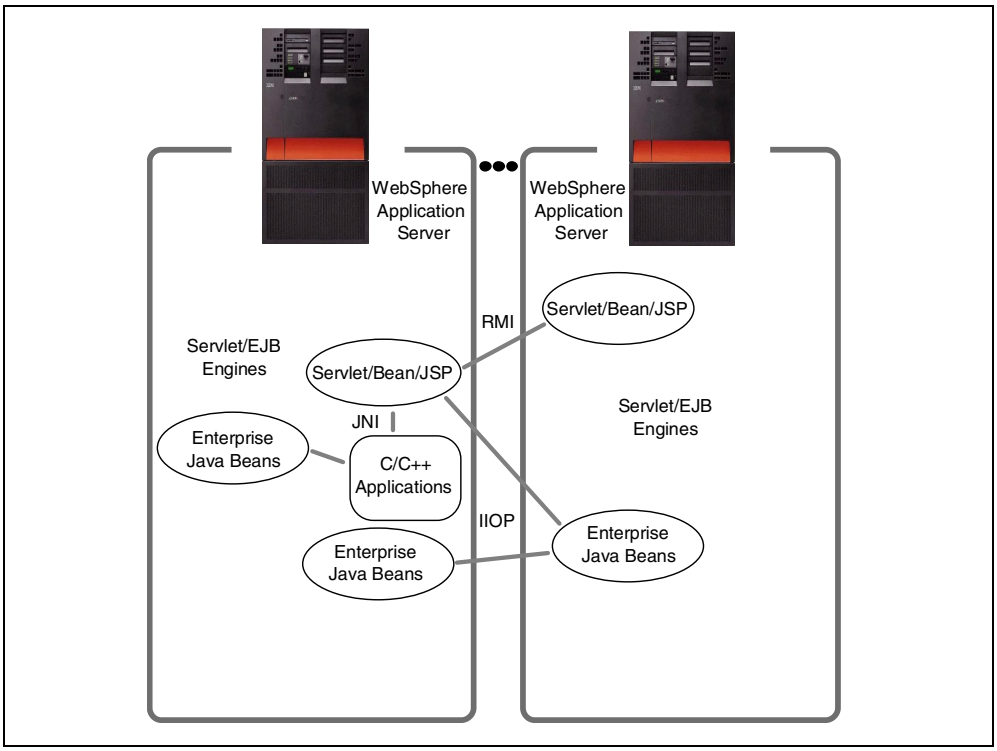


Figure 126. Middle tier: Application server

The back tier can be on the same physical machine as the application server, or on any number of separate servers containing the business logic and data. In WebSphere, gateway connectors are Java-based applications that convert HTML requests into sets of parameters that are passed to a client process on the application server. These client processes communicate with the back tier application and database servers using native or Java-based APIs. Similarly, adapter connectors provide a direct mapping to applications and data on the back tier systems and provides a transactional framework across disparate systems (Figure 127). Gateway connectors are easier to develop than adapter connectors. However, the adapter connectors are more scalable and perform better.

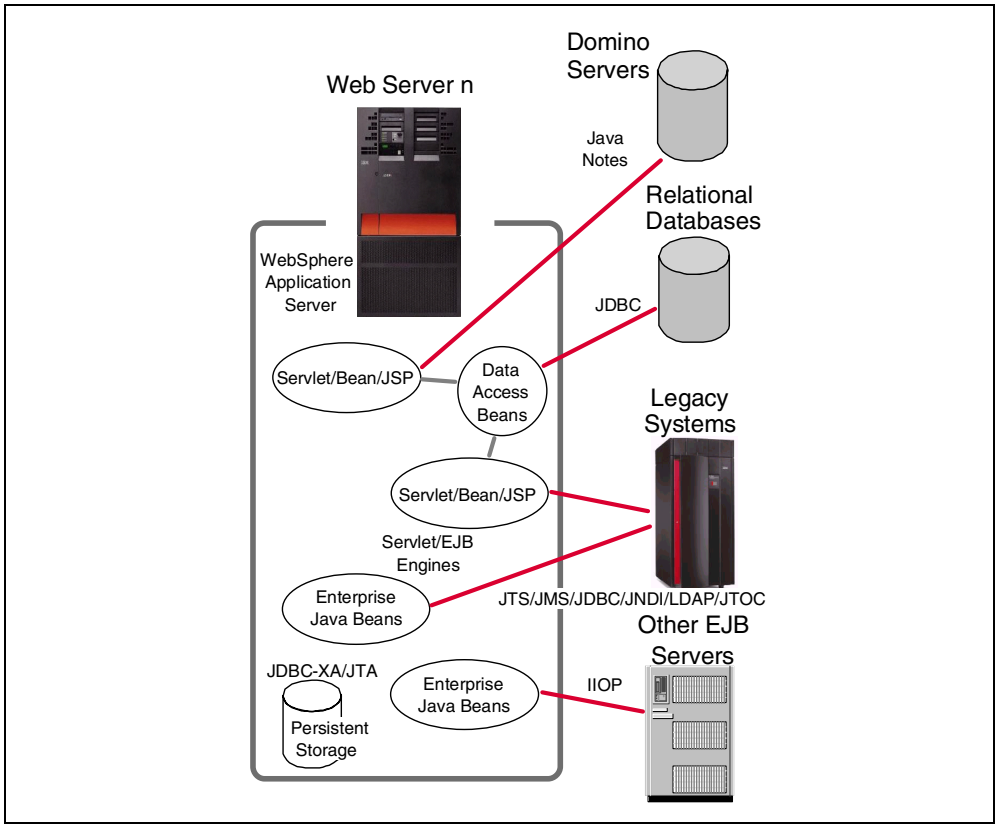


Figure 127. Back tier connectivity

From a performance sizing and capacity planning perspective, this can range from moderately simple to quite complex:

- Servlets or Java server pages with minimal processing and no database access and no connections to another back tier server, should map to the hits/second/CPW metric listed earlier.
- Servlets or Java server pages with more extensive processing and database reads or writes, and no connections to another back tier server, should map to a lower hits/second/CPW metric than for static pages.
- Servlets or Java server pages accessing back tier servers will be subjected to a certain hits/second/CPW metric. Back tier servers will be subjected to transactions requiring CPU, disk, and main memory resources. The middle tier servers and back tier servers communication IOP will incur a load. There will also be latency introduced by the network connection.

There are still contributions from the client, network, and server resources. However, the contributions are now associated with the back tier server and the network connection between the application server and back tier servers.

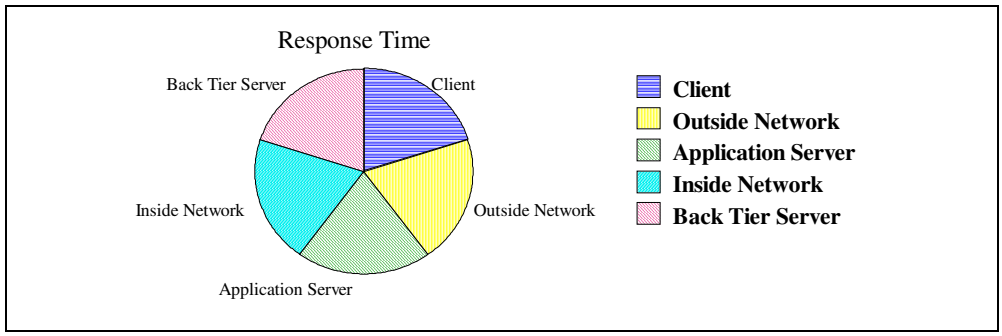


Figure 128. Response time components for three-tier application server architecture

Appendix B. Getting more detailed assistance

In many cases, your own intuition, plus the tools and techniques described in this publication, will enable you to accurately and efficiently build and deliver Web-based solutions. However, complex applications or networking concepts may force you to seek outside help. There are a number of IBM organizations that can provide assistance, some are free and some are fee-based.

B.1 AS/400 Benchmark Centers

The AS/400 Customer Benchmark Centers in Rochester, MN and Santa Palomba, Italy are part of the Partners in Development organization. They are available to help you whenever system performance measurements are required. The centers can provide custom batch, interactive, and client/server benchmark services on custom AS/400 hardware configurations. They can also support remote dial-in to Rochester for batch benchmarks. Services are provided on a cost recovery bases. For more detailed information, or to obtain a nomination form, visit their Web site at: `http://www.as400.ibm.com/developer/cbc/index.html`

B.2 IBM International Networking Center

The IBM International Networking Center offers highly effective programs to help you create a strategy that addresses your business needs, plan for changes or migrations, and develop or review a network or management design. Using experts with extensive consulting and design experience, the Center provides guidance to worldwide clients with networking, management, network computing, and e-business issues. For more detailed information, visit their Web site at:
`http://www.networking.ibm.com/ntc/ntcover.htm`

B.3 IBM Solution Partnership Centers (SPC)

The mission of the SPC is to help developers port, enable and market their applications on IBM hardware and software platforms. SPCs are comprised of professionally staffed porting labs with a variety of hardware, software, and middleware pre-configured to your specifications. An Internet-specific porting center, expert technical assistance and education, business seminars, marketing support, and the latest information about industry and technology directions are available. For more detailed information, visit their Web site at:
`http://www.developer.ibm.com/spc/index.html`

B.4 IBM Global Services

IBM Global Services can provide a number of helpful services to assist you in capitializing on the potential of e-business. Whether your need is business process reengineering, complex information technology system design, application development, or even secure hosted application services, IBM Global Services can help. For more detailed information, visit their site on the Web at:
`http://www.ibm.com/services/e-business/`

Appendix C. Web serving performance measurements

Table 30 provides a summary of the measured performance data. This table should be used in conjunction with the rest of the information in this redbook for correct interpretation. Results listed here do not represent any particular customer environment. Actual performance may vary significantly from what is provided here.

Table 30. Comparision table for AS/400 on V4R4M0

Hits per Second per CPW

Transaction Type	Non-secure				Secure			
	V4R2 Token Ring	V4R3 Token Ring	V4R4 Token Ring	V4R4 Ethernet	V4R2 Token Ring	V4R3 Token RIng	V4R4 Token Ring	V4R4 Ethernet
Static Page - not cached	0.656	1.110	1.110	1.180	0.290	0.450	0.450	0.480
Static Page - cached*	N/A	1.560	1.840	1.860	N/A	0.520	0.530	0.560
CGI (HTML) - New Activation	0.106	0.070	0.070	0.070	0.087	0.060	0.060	0.060
CGI (HTML) - Named Activation	NA	0.250	0.350	0.440	0.087	0.190	0.240	0.280
CGI (HTML) - Persistent	NA	0.260	0.370	0.440	NA	0.180	0.230	0.250
CGI (SQL) - New Activation	0.069	0.060	0.050	0.060	0.59	0.060	0.050	0.060
CGI (SQL) - Named Activation	NA	0.250	0.340	0.430	0.59	0.190	0.230	0.280
Net.Data (HTML)	0.086	0.140	0.220	0.240	0.072	0.120	0.160	0.190
Net.Data (SQL)	0.047	0.120	0.140	0.150	0.039	0.100	0.120	0.130
Net.Commerce - cached	NA	0.250	NA	0.250	NA	NA	NA	NA
Net.Commerce - not cached	NA	0.006	NA	NA	NA	NA	NA	NA
Java Servlet	NA	NA	NA	0.400	NA	NA	NA	0.280

1000 Character Pages

*** Examples:**

- **o V4R3 S30-2257 (CPW=310): 319 x 1.56 = 497+ hits/second, 1,789,200 hits/hour.**
- **o V4R4 S30-2257 (CPW=310): 319 x 1.84 = 586+ hits/second, 2,109,600 hits/hour**
- **o V4R4 S30-2257 (CPW=310): 319 x 1.88 = 599+ hits/second, 2,156,400 hits/hour**

NA = Not Available

Appendix D. Special notices

This publication is intended to help Customers, IBM Business Partners and IBM Specialists to analyze, tune, and size Internet applications for the AS/400. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/400 and the AS/400 Performance Tools. See the PUBLICATIONS section of the IBM Programming Announcement for AS/400 Performance Tools for more information about what publications are considered to be product documentation.

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.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

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 Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including, in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

AIX
AS/400
AT
CICS
CT
DB2
IBM®
IBM Payment Servers
Net.Data
Netfinity
Nways
OS/400
RS/6000
SecureWay
SP
System/390
WebSphere
XT
400

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

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

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Appendix E. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 IBM Redbooks publications

For information on ordering these ITSO publications see “How to get IBM Redbooks” on page 201.

- *AS/400 Internet Security: IBM Firewall for AS/400*, SG24-2162
- *Safe Surfing: How to Build a Secure WWW Connection*, SG24-4564
- *AS/400 Performance Management, V3R6/V3R7,* SG24-4735
- *Cool Title About the AS/400 and Internet,* SG24-4815
- *AS/400 Internet Security: Protecting Your AS/400 from HARM in the Internet*, SG24-4929
- *Unleashing AS/400 Applications on the Internet,* SG24-4935
- *Lotus Domino for AS/400: Performance, Tuning, and Capacity Planning,* SG24-5162
- *Net.Commerce V3.2 for AS/400: A Case Study for Doing Business in the New Millennium*, SG24-5198
- *A Comprehensive Guide to Virtual Private Networks, Volume I: IBM Firewall, Server and Client Solutions*, SG24-5201
- *Internet Security in the Network Computing Framework*, SG24-5220
- *A Comprehensive Guide to Virtual Private Networks, Volume II: IBM Nways Router Solutions*, SG24-5234
- *A Comprehensive Guide to Virtual Private Networks, Volume III: Cross-Platform Key and Policy Management*, SG24-5309
- *IBM Firewall for AS/400 V4R3: VPN and NAT Support*, SG24-5376
- *TCP/IP Tutorial and Technical Overview*, GG24-3376

The following publications are available in softcopy only on the Web at:
`http://www.redbooks.ibm.com`

- *AS/400 Server Capacity Planning,* SG24-2159
- *AS/400 Performance Explorer Tips and Techniques,* SG24-4781
- *AS/400 Communication Performance Investigation - V3R6/V3R7*, SG24-4895
- *An Implementation Guide for AS/400 Security and Auditing: Including C2, Cryptography, Communications, and PC Connectivity*, GG24-4200

At the site, click **Redbooks Online!**. In the search engine field, type the publication number, and click **Submit Search**. Click the matching publication title that appears to view the document.

E.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

E.3 Other resources

These publications are also relevant as further information sources:

- *Performance Tools/400, V3R7*, SC41-4340
- *AS/400 Tips and Tools for Securing Your AS/400 V4R4*, SC41-5300
- *AS/400 Security - Basic V4R1*, SC41-5301
- *OS/400 Security - Reference V4R4*, SC41-5302
- *AS/400 Security - Enabling for C2*, SC41-5303
- *AS/400 Work Management V4R4,* SC41-5306
- *Performance Tools V4R2,* SC41-5340
- *BEST/1 Capacity Planning Tool for V4R1,* SC41-5341
- *TCP/IP Configuration and Reference,* SC41-5420
- *DB2 for AS/400 SQL Programming Version 4*, SC41-5611
- *HTTP Server for AS/400 Webmaster's Guide V4R4,* GC41-5434

The following publications are available in softcopy only on the Web at:
http://as400bks.rochester.ibm.com/pubs/html/as400/onlinelib.htm

- *Web Programming Guide V4R4*, GC41-5435.
- *AS/400 Performance Capabilities Reference V4R4,* SC41-0607

At the site, select your language preference, and press **Go**. Select **V4R4**. Then, select **Search or view all V4R4 books**. Enter the publication number in the search field and click **Find.** Then, click the appropriate title.

E.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- An extensive AS/400 library and download center that includes AS/400 service (PTFs, downloads, and WebSphere forum), Technical Studio, Web builders workshop, and Domino for AS/400:
  http://www.as400.ibm.com/support/
- The AS/400 Information Center is located at:
  http://publib.boulder.ibm.com/pubs/html/as400/infocenter.html/ and
  http://www.as400.ibm.com/infocenter/
- The IBM AS/400 Web site: http://www.as400.ibm.com/
- The IBM product publications Web site: http://as400bks.rochester.ibm.com/
- The IBM Partners In Development Web site:
  http://www.as400.ibm.com/developer/index.html
- The IBM Global Services Web site: http://www.ibm.com/services/ or
  http://www.ibm.com/services/e-business/
- The WebTrends Analyzer, from WebTrends Corporation in Portland, Oregon, can be found at this site: http://www.webtrends.com/
- NetIntellect, from WebManage Technologies, Inc., can be found at this site:
  http://www.webmanage.com/
- Many publications cited in this redbook can also be found at this site:
  http://as400bks.rochester.ibm.com/pubs/html/as400/onlinelib.htm/

  At the site, select your language preference, and press **Go**. Select the relevant version and release (such as, **V4R4**). Then, select **Search or view all VxRx books**. Enter the publication title or number in the search field and scroll down until you find the appropriate title.
- For good disk arm sizing recommendations and requirements based on processor model performance, go to:
  http://www.as400.ibm.com/developer/performance/as4armct.html/
- For a variety of networking case studies, including Internet, Ethernet, and ATM solutions, go to: http://www.networking.ibm.com/case/studies.html/
- Authorized IBM business partners can download the AS/400 Workload Estimator from this site (user ID and password required):
  http://partners.boulder.ibm.com/
- The PM/400e Performance Management Service site is at:
  http://www.as400.ibm.com/pm400/pmhome.htm/
- *AS/400 Magazine* Web site: http://www.as400magazine.com/
- Use this search engine to find specific AS/400 information:
  http://www.search400.com/
- For an overview of IBM Network Dispatcher products, go to:
  http://www.networking.ibm.com/white/serverload.html/
- For a general overview of Internet Traffic Management, go to:
  http://www.itmcenter.com/
- The DB2/400 technical tips and techniques Web page:
  http://www.as400.ibm.com/db2/db2tch_m.htm/

- The AS/400 Benchmark Center home page:
  http://www.as400.ibm.com/developer/cbc/index.html/
- The IBM Solution Partnership Centers (SPC) home page:
  http://www.developer.ibm.com/spc/index.html
- The IBM International Networking Center home page:
  http://www.networking.ibm.com/ntc/ntcover.htm

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number

☐ Credit card number

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Q

R

S

T

U

V

W

IBM Redbooks evaluation

AS/400 HTTP Server Performance and Capacity Planning
SG24-5645-00

Your feedback is very important to help us maintain the quality of IBM Redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

_____

_____

_____

_____

What other Redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

_____

SG24-5645-00

Printed in the U.S.A.

AS/400 HTTP Server Performance and Capacity Planning

SG24-5645-00