WHITE PAPER

July 1997

Prepared By
Intranet Solutions
Business Unit

Compaq Computer
Corporation

CONTENTS

COMPAQ

529A/0697

Performance Analysis and Capacity Planning of Netscape Enterprise Server 2.01 on Compaq Servers

Web technologies have exploded in recent years. More businesses and organizations are experimenting with Internet technologies, due in large part to the benefits facilitated by them, including the ability to quickly develop and deploy applications within diverse environments, and implementation of seamless communications. Performance issues have therefore become critical topics in Internet/intranet design. Good performance analysis can help users to:

- *Select appropriate hardware and software in Internet/intranet design*
- *Facilitate capacity planning*
- *Make configuration adjustments to achieve maximum output*
- *Identify the bottlenecks in a system*
- *Understand user service loads*

Internet/intranet performance analysis is complicated, since the performance is related to a variety of components, such as network, web server software, and hardware. Carefully selecting correct methods to do the performance analysis is very important.

NOTICE

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state or local requirements. Compaq does not warrant products other than its own strictly as stated in Compaq product warranties.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Compaq, Compaq Insight Manager, ProLiant, NetFlex, registered United States Patent and Trademark Office.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

Microsoft, Windows, Windows NT, Windows NT Advanced Server, SQL Server for Windows NT are trademarks and/or registered trademarks of Microsoft Corporation.

Netscape, Enterprise Server are trademarks and/or registered trademarks of Netscape Communications Corp.

Performance Analysis and Capacity Planning of Netscape Enterprise Server 2.01 on Compaq Servers

First Edition (July 1997)
Document Number 529A/0697

PERFORMANCE METRICS

Internet/intranet performance can be viewed in either of two ways: by a Network Administrator or Web Master concerned with system throughput and utilization, or by the individual end user concerned with response time.

This paper focuses on the performance of Netscape Enterprise Server 2.01 running on Compaq servers. Requests per second and latency time (as described below) were selected as the web performance metrics.

- *Requests per second* - The total number of requests per second completed during specified test period.
- *Latency time* - Amount of time that a request for a client takes. Latency includes both the time it takes a client to connect to the server and the time it takes for the server to transfer data to the client. Latency is expressed in milliseconds in this paper.

Performance Analysis Methodologies

For the performance analysis investigation, Compaq used three methods: modeling user requirements, benchmarking tests, and monitoring performance.

Modeling User Requirements

Generally, the contents of a web site can be divided into two categories: static and dynamic. Three sample user profile models are presented here. The first is the Static HTML model. Depending on whether the files are HTML or graphic, all the requests are static HTTP GET requests. The analysis shows more than 90% of sites belong to this model. The second is the Light CGI model. About 10% of the requests will be CGI, and 90% are static. The third is the Heavy CGI model. About 90% of the requests will be CGI, and 10% are static. The three models are illustrated in Table 1.

TABLE 1. MODEL OF USER REQUIREMENT

User Models:	CGI Requests:	Static HTML Requests:
Static HTML	0%	100%
Light CGI	10%	90%
Heavy CGI	90%	10%

Benchmarking Tests

Benchmarking tests are generally used in web server performance analysis. The standard benchmark tools provide the ability to run exactly the same test scenario under different configurations. The configurations include both hardware and software, such as different operating systems and web servers, so only certain parameters are changed to isolate their effects on performance.

There are currently several web server performance benchmark tools available. WebStone, SPECweb, and WebBench are the most commonly used benchmark tools.

- WebStone - WebStone measures the raw throughput of a standard HTTP workload. It is measured across two main variables - latency in seconds and the number of connections per second.

- SPECweb - developed by the SPEC committee, measures the response time for server requests across a number of different workloads. It sends HTTP requests to the server, based on defined workload parameters, and calculates the overall throughput at the end of a run.
- WebBench - developed by Ziff-Davis Benchmark Operation. WebBench provides both static standard tests and dynamic tests. The dynamic tests measure access to the Common Gateway Interface scripts. In the general reports of WebBench, there are two overall server scores: request per second and throughput, as measured in bytes per second. The latency time is generated from client reports.

In this paper, all the benchmarking results were gathered using WebBench 1.0.

Monitoring Performance

During benchmarking tests, users are also interested in performance on each subsystem, such as CPU, memory, disk, and network. Compaq used PerfMon and NetMon to monitor the behavior of each subsystem, which is measured by CPU utilization, memory usage, and network utilization.

Monitoring performance is very helpful in addressing the bottlenecks in the systems.

PERFORMANCE TEST SETUP

Compaq used WebBench 1.0 to implement the benchmarking performance tests. The tests were designed to show the performance of Compaq servers running Netscape Enterprise Server 2.01 on Microsoft Windows NT 4.0.

Benchmarking Test Setup

The WebBench 1.0 setup includes two aspects: the suite setup and the test environment setup.

Suite Setup

The test suites reflect the user requirement. Three test suites were created to correspond to the three selected user models. For the Static Test suite, the workload is 100% static requests. For the Light CGI test suite, the distribution is 90% static requests and 10% dynamic requests. The Heavy CGI test suite contains 10% static requests and 90% CGI dynamic requests.

In each test suite, several tests were run. Figure 1 shows the Light CGI suite edit screen.

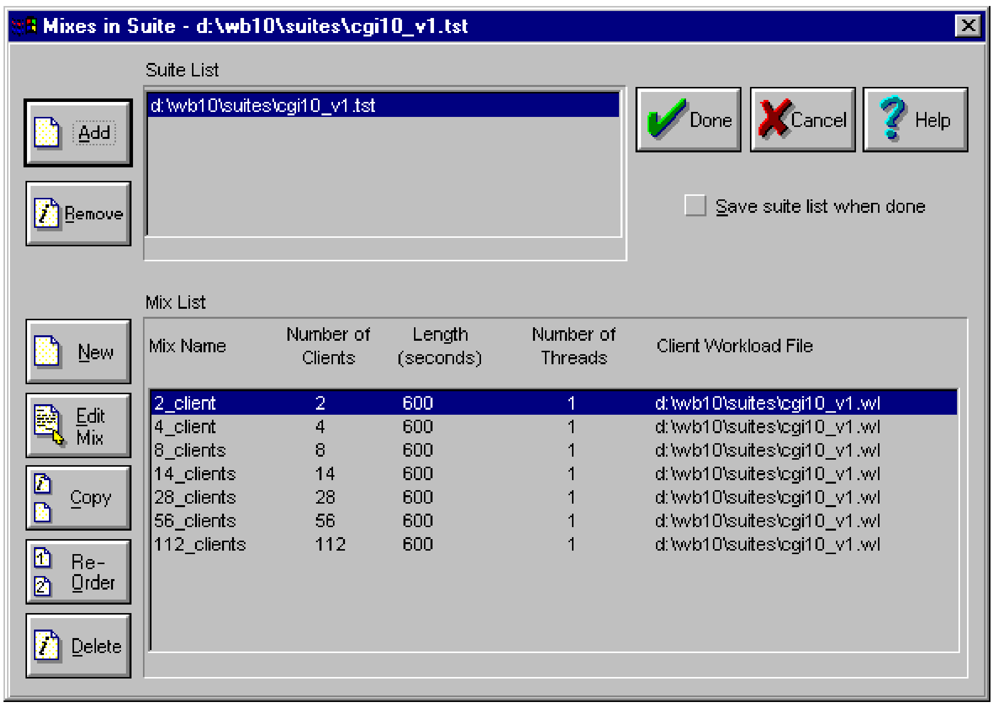


Figure 1: Light CGI suite edit screen

The virtual client numbers were changed to simulate different loads. Other testing parameters were unchanged to isolate the loading effects on the tested server. These parameters included:

- Running time of 10 minutes
- Ramp up and ramp down time of 30 seconds each
- Thread number of each client was one
- Receive buffer size of each client was 4K
- Keep-alive turned off
- Thinking and delay time were zero

Figure 2 shows the test parameter edit screen.

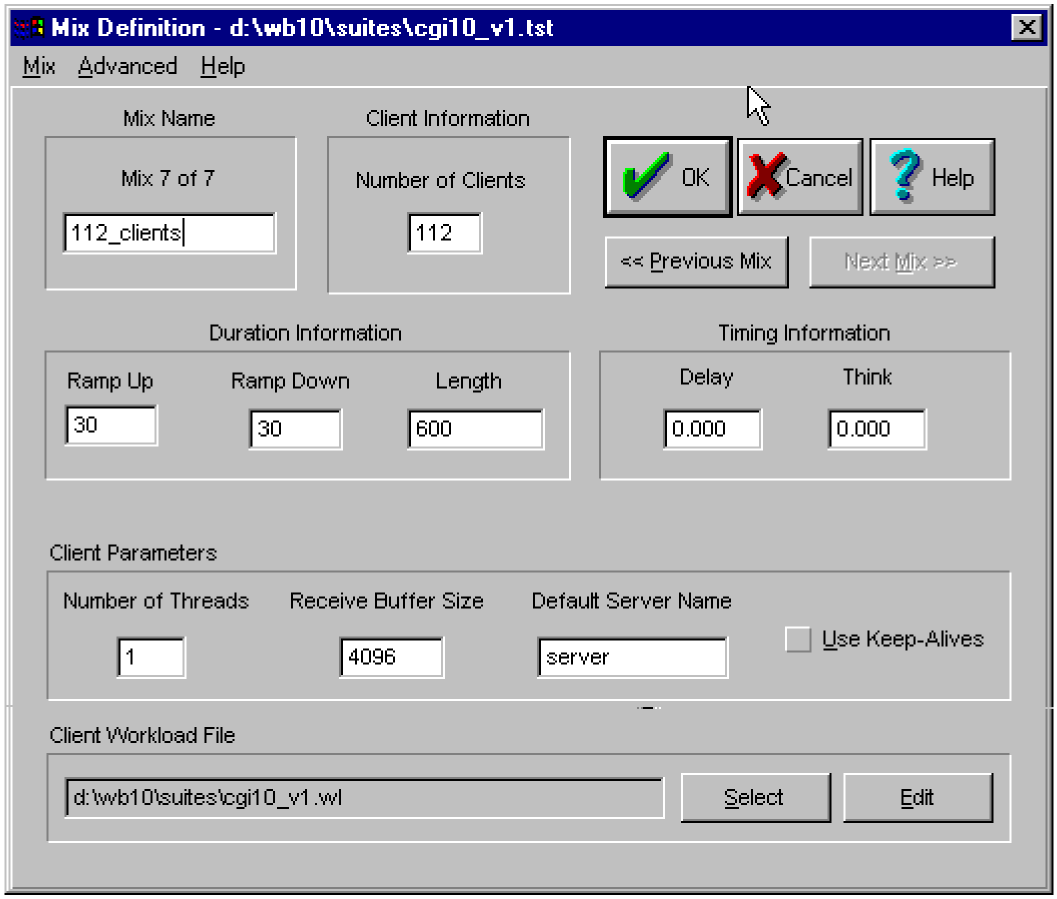


Figure 2: Testing parameter editing screen

Testing Environment Setup

Figure 3 shows an overview of the testing environment setup for the static suite test in the testing lab at Compaq. The controller is a ProLiant 4500 with one Pentium Pro processor and 256 MB of RAM. The controller sends the test information to the clients, and when the clients complete the tests, the results are sent back to the controller.

The clients send HTTP requests to the tested server and collect data. In static suite tests, 24 client machines were deployed in two dedicated 100-Mbit fast Ethernet LANs. The test servers have two network interface cards to connect the two subnets. These two 100-Mbit NICs in the test server push it near its network performance limit.

Compared to static suite tests, 100-Mbit network bandwidth is wide enough for the Light CGI and Heavy CGI tests. One dedicated 100-Mbit LAN was used in the CGI tests, and the number of client machines was reduced to 14, as 14 machines could generate enough loading in CGI tests for the tested servers.

The number of clients was increased during the tests to simulate different work loads. When the number of clients was bigger than the number of client machines, virtual clients participated.

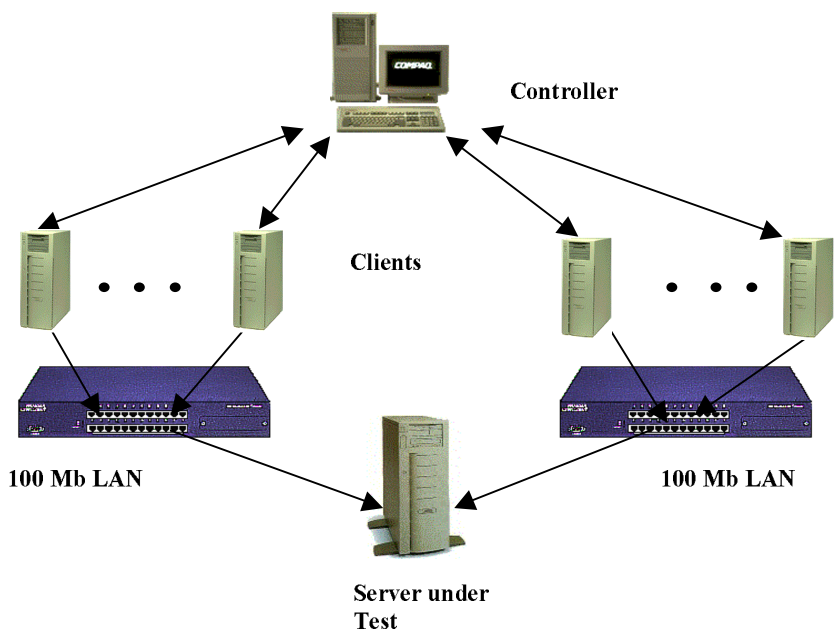


Figure 3: Overview of Testing Environment

Tested Server Setup

Since WebBench measures the performance of the Web server software and hardware, the configuration of the operating system and Netscape Enterprise Server were kept the same. Compaq servers and their configurations were changed during the performance tests to isolate the effects on performance based on hardware and related configurations.

TABLE 2. CONFIGURATION OF OS AND WEB SERVER SOFTWARE

Web Server Software:	**Vendor**	Netscape Communications Corp.
	HTTP Software	Enterprise Server 2.01
	Server Cache	Default
	Log Mode	Common
	Number of threads	Default
Operating System:	**Vendor**	Microsoft
	Version	NT 4.0
	Others	Service Pack 2 with the tcpip.sys updated file

The Compaq ProLiant 800 was selected as the baseline server and the Compaq ProLiant 5000 was selected as the high-end server. Table 3 lists the different testing configurations of the two servers. All of them were tested using the three user requirement models: Static, Light CGI, and Heavy CGI.

TABLE 3. LIST OF HARDWARE CONFIGURATIONS

Machine:	Processors:	Memory:
ProLiant 800	1 Pentium Pro 200/256K Cache	64 MB
ProLiant 800	2 Pentium Pro 200/256K Cache	64 MB
ProLiant 5000	1 Pentium Pro 200/512K Cache	256 MB
ProLiant 5000	1 Pentium Pro 200/512K Cache	512 MB
ProLiant 5000	2 Pentium Pro 200/512K Cache	256 MB
ProLiant 5000	2 Pentium Pro 200/512K Cache	512 MB

RESULTS ANALYSIS

In March, 1997, Compaq conducted a series of benchmarking performance tests on the ProLiant 800 and ProLiant 5000 servers running Netscape Enterprise Server 2.01. While using benchmark programs to gather the performance data, monitoring tools like PerfMon and NetMon were used to view other parameters such as CPU utilization, network utilization, and memory usage. The following section analyzes the results based on different user requirement models and different Compaq servers.

Benchmarking tests were conducted on six hardware configurations using three user-load models. The ProLiant 5000 and ProLiant 800 both showed good scalability from 1 processor to 2 processors. Compared with the Static Model, the throughput of servers was less in the CGI models. This reduction in throughput is caused by CGI requests using more server system resources, such as CPU time, file handles, and so on.

Server performance was not affected by changing the memory configuration in any of the three models.

Static Model

In static model benchmarking tests, all requests use the static GET method. The construction of sample files is based on the following rules:

- Average file size is 8 KB
- 92% of files are less than 32 KB
- The size of the most frequently retrieved file is 4 KB
- The smallest file is 256 bytes
- The largest file is 128KB

Exact distribution of the sample files can be found in Appendix A.

Previous testing showed that it was very easy for the ProLiant 800 and ProLiant 5000 to saturate one 100-Mbit Fast Ethernet LAN card in static model benchmarking tests. Therefore, two dedicated 100-Mbit Fast Ethernet LAN cards were used in the server under tests, so that the tests could stress the Compaq servers. Both ProLiant 800 and ProLiant 5000 servers can be configured with more than two Network Interface Cards. Since testing also showed that a client machine cannot accommodate more than 10 virtual clients, 24 client machines were used to generate loads.

Static Model tests showed:

- The benchmarking tests stressed the servers, evidenced by the CPU utilization reaching 95%-100% in peak performance situations for all six hardware configurations.
- The ProLiant 5000 with 2 processors can handle as many static HTTP requests as the two dedicated 100-Mbit LANs can send.
- Both ProLiant 5000 and ProLiant 800 showed good scalability from 1 processor to 2 processors.
- Performance increased very quickly as the client number increased, then held steady. The servers showed the ability to handle many clients simultaneously, without reducing the throughput of the servers.

ProLiant 5000

Benchmarking Results

Figures 4 and 5 show the benchmarking results of a ProLiant 5000 system with 1 Pentium Pro 200-MHz processor, 512K L2 on-chip cache, 256-MB RAM, and 2 PCI NetFlex-3 10/100 NICs. The web server and operating system configuration can be found in Table 2.

The throughput number jumped from 366 requests/sec on 12 virtual clients to 988 requests/sec on 48 virtual clients as the loading was increased. The throughput number increased slowly to 1072 requests/sec on 96 virtual clients. After that, the number drops but still keeps 900 requests/sec to 1000 requests/sec.

Latency does not show a big increase from 12 virtual clients to 24 virtual clients, which are 31 milliseconds and 34 milliseconds respectively. The number stayed linear, increasing from 48 virtual clients to 216 virtual clients.

No errors occurred in the tests.

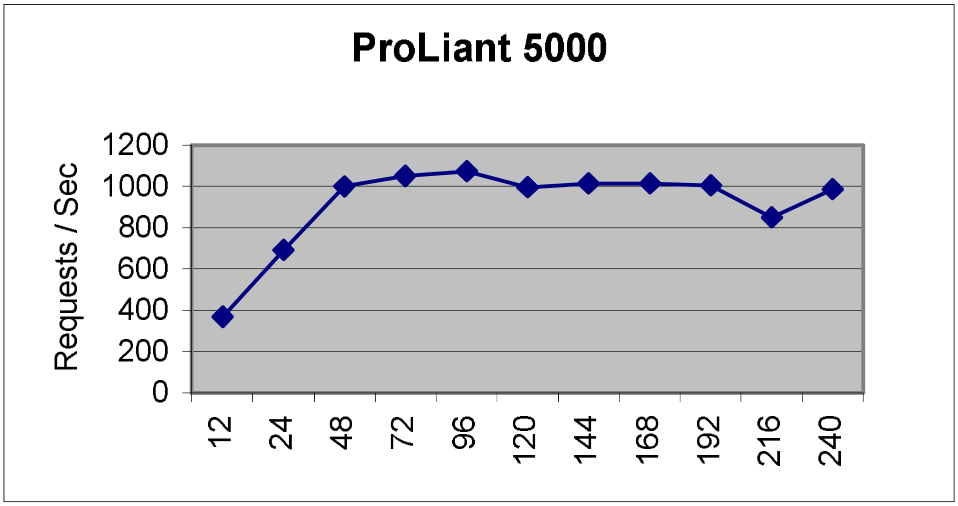


Figure 4: Throughput for ProLiant 5000 1P with 256-MB RAM

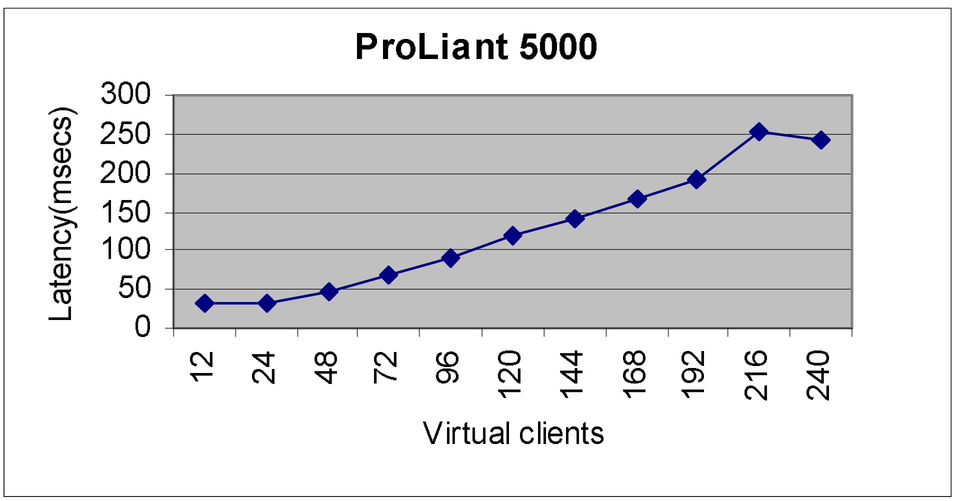


Figure 5: Latency for ProLiant 5000 1P with 256-MB RAM

Processors

A Pentium Pro 200-MHz processor with 512K L2 on-chip cache was added to the previous system. Other subsystems were kept the same, as well as the configuration of the operating system and web server software. Figures 6 and 7 clearly show the performance increase when moving from a 1 processor system to a 2-processor system. When comparing the two setups for peak performance, the 2-processor system exhibited a 30% advantage.

The throughput did not change when the number of virtual client numbers increased from 12 and 24. However, the second processor caused a decrease of 2- 4% latency in the low-loading situation. When the virtual client number was over 48, the second processor showed a big advantage. The 1-processor system showed a performance drop after more than 96 virtual clients, compared to the 2-processor system that showed performance consistency with over 1350 requests per second, even when there were 240 virtual clients.

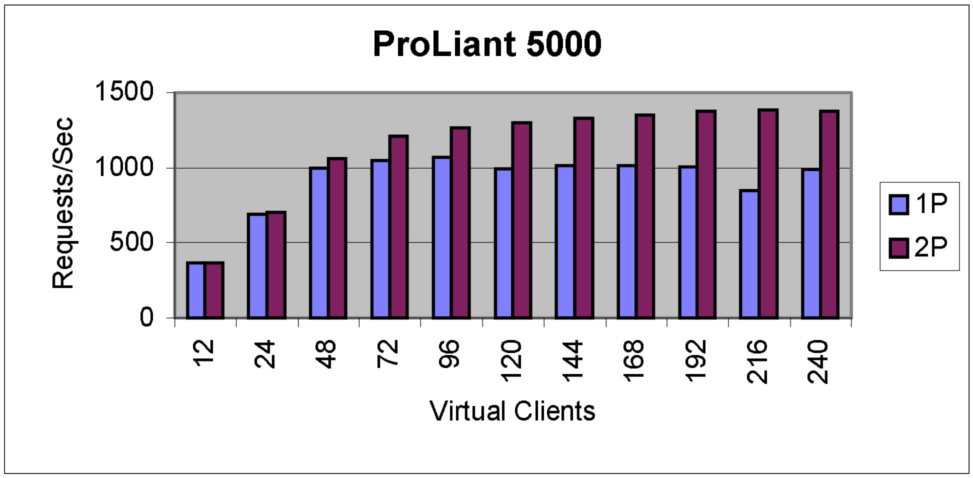
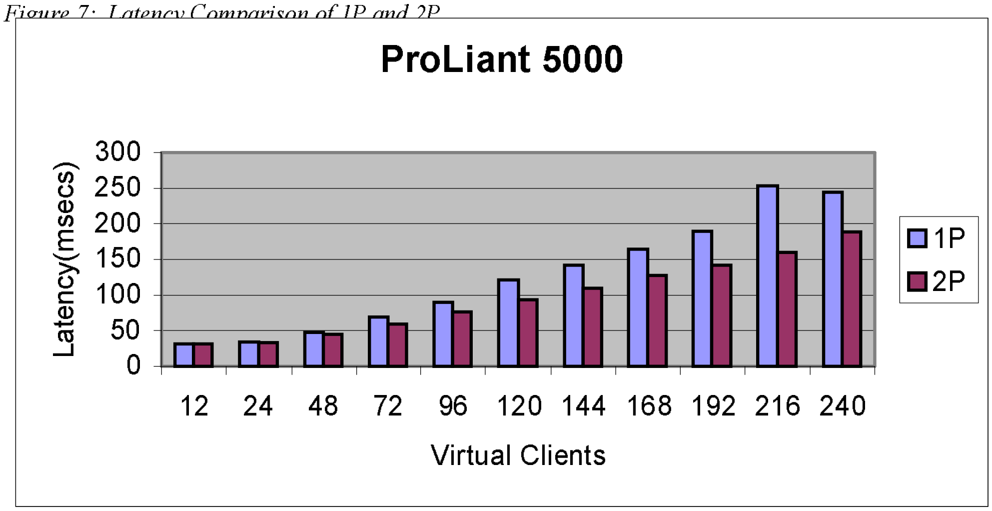


Figure 6: Throughput Comparison of 1P and 2P

Figure 7: Latency Comparison of 1P and 2P



Memory

Memory configurations of the 1-processor and 2-processor systems were changed from 256-MB to 512-MB. Other configurations were not changed. Figures 8 and 9 show the performance comparison after the memory changes. For both 1P and 2P systems, adding more memory only showed a slight performance increase. This is attributed to the fact that the Netscape Enterprise Server can cache the static files, and there was a relatively small set of sample files. Even in a small memory situation, the server could cache all the sample files.

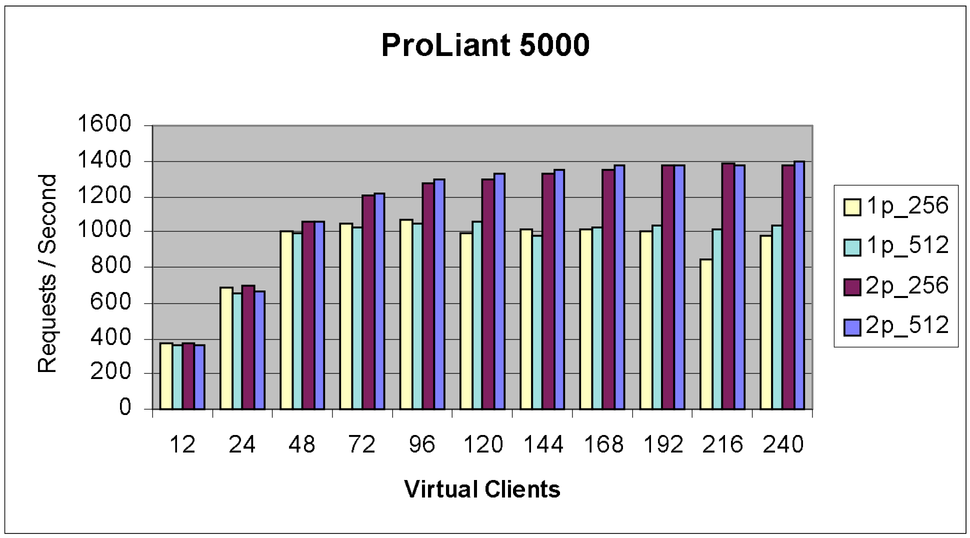


Figure 8: Throughput Comparison on Different Memory Configurations

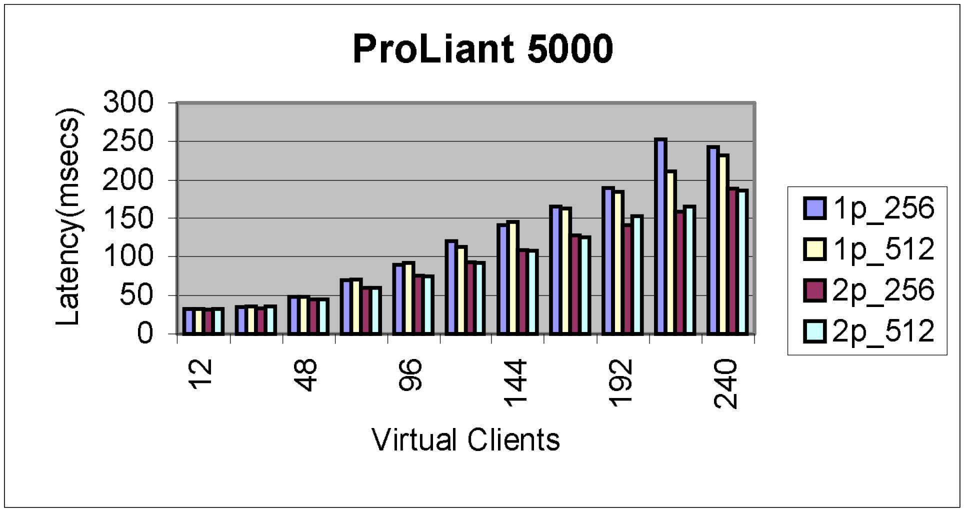


Figure 9: Latency Comparison of Different Memory Configurations

Monitoring Results

During the benchmarking performance tests, three characteristics were monitored: CPU utilization, memory usage, and network utilization. We viewed the network utilization from the hubs, and the CPU utilization and memory usage from the Windows NT Performance Monitor program. To limit the monitoring program's effect on performance, the data was not recorded to disk. The results were used to indicate the system condition of the test servers.

Table 4 shows the monitoring results of different hardware configurations for the static model at peak performance situations.

TABLE 4. MONITORING RESULTS FOR STATIC MODEL AT PEAK PERFORMANCE SITUATIONS

Hardware Models:	Client Number:	CPU Utilization:	Network Utilization:	Memory Usage:
ProLiant 5000 1P	96	95%~100%	35%~50%	60MB~65MB
ProLiant 5000 1P+	96	95%~100%	35%~50%	60MB~65MB
ProLiant 5000 2P	240	About 95%	50%~75%	60MB~70MB
ProLiant 5000 2P+	240	About 95%	50%~75%	60MB~70MB

ProLiant 800

The ProLiant 800 server includes either one or two 200-MHz Pentium Pro Processors with integrated 256-KB, second-level cache. Two Netflex-3 PCI NICs were added to the test server. Both servers had 64-MB RAM.

The 2-processor ProLiant 800 reached about 1000 requests/sec when there were 48 clients. Its throughput remained steady at about 1000 requests/sec for the rest of the tests. The peak performance was reached at 1031 requests/sec on 216 virtual clients. The 1-processor ProLiant 800 reached 742 requests/sec on 120 virtual clients at its peak performance.

The ProLiant 800 showed similar performance trends with the ProLiant 5000 (as seen in Figures 10 and 11). In peak benchmarking performance results, the 2-processor ProLiant 800 had about a 39% performance increase over the 1-processor ProLiant 800.

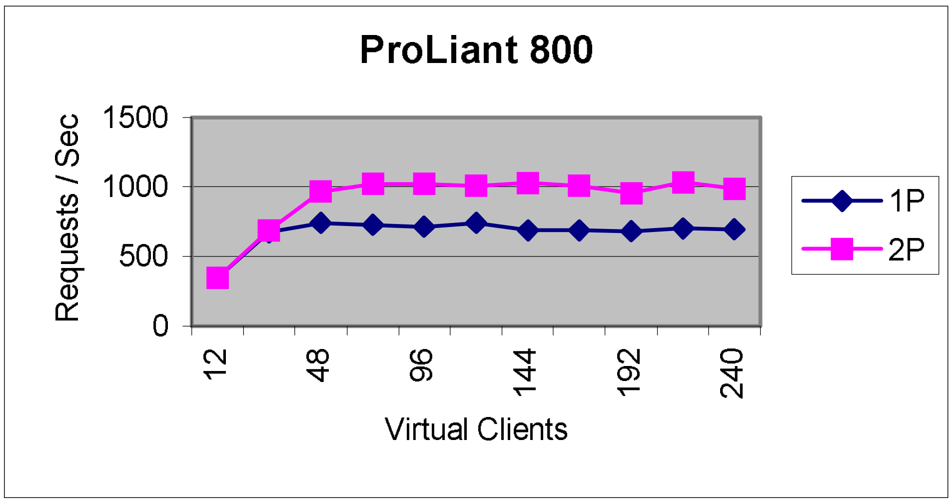


Figure 10: Throughput Comparison for ProLiant 800 1P and 2P

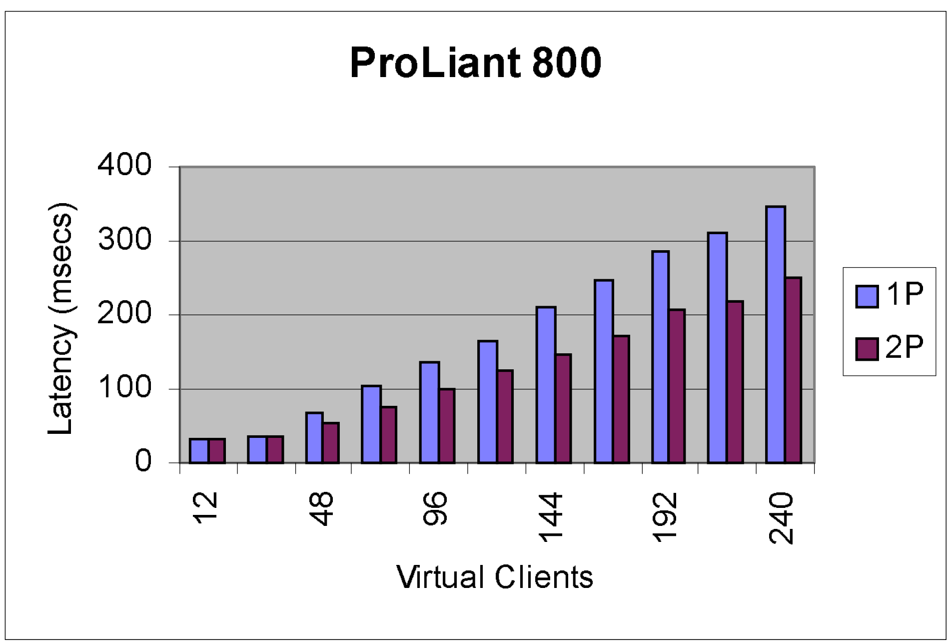


Figure 11: Latency Comparison of ProLiant 800 1P and 2P

Light CGI Model

The Light CGI model consists of 10% dynamic CGI requests and 90% static requests. The dynamic CGI request used in the benchmark test invokes a program in the server side to create a text file containing the system variables, parameters, and values. The file size is dependent on the server condition, and is normally around 670 bytes. The sample output is listed in Appendix B.

The setup used one dedicated 100-Mbit Fast Ethernet card in the Light CGI model benchmarking tests, as the server always gets saturated in the CGI tests. The operating system and web server software configurations were the same as those for the static model.

The CGI model tests showed:

- The servers became saturated as the CPU utilization reached near 100% in a peak-performance situation.
- Compared to the Static model, throughput is reduced. CGI requests consume more server system resources, such as CPU time, file handles, and so on.
- The latency increased in CGI models. When servers receive a dynamic CGI request, a server side program is invoked, which consumes some CPU time. It takes a much longer time for the server to process CGI requests than static requests.
- Both the ProLiant 5000 and ProLiant 800 show good scalability from 1 processor to 2 processors in CGI-model benchmarking tests.

ProLiant 5000

Figure 12 shows the benchmarking results of different configurations of the ProLiant 5000 in the Light CGI model. There are four setups listed in the graph: 1-processor Pentium Pro 200 with 512K cache processor with 256-MB RAM, then with 512-MB RAM, and the 2-processor Pentium Pro 200 with 256K L2 cache with 256 MB RAM, then with 512-MB RAM. In this chart the 2-processor server showed a big advantage over the 1-processor server in both the 256 MB and 512MB memory categories when there were more than 14 virtual clients. The 2-processor server produces about 70% more throughput than the 1-processor server in these tests.

As with the static tests, adding more memory to the 1-processor or 2-processor server did not show performance improvement in the benchmarking tests for the Light CGI model.

The highest throughput for the 1P ProLiant 5000 was 215 requests/sec and 344 requests/sec for the 2P ProLiant 5000. These were reached at 28 and 56 virtual clients, respectively.

There were two stages for the benchmarking tests. The first stage for 1 processor was from 2 clients to 8 clients. In this first stage, the throughput increased significantly when the client number increased; however, latency increased very slowly. The clients didn't generate enough load to stress the tested server. The first stage for 2 processors was from 2 clients to 14 clients, and lasted a little bit longer than the first stage for 1 processor, since the 2 processor handles more requests.

The second stage for 1 processor was from 14 clients to 112 clients. When the client number increased, the throughput remained steady and the latency increased. This indicated that the load was high enough to stress the servers. The servers handled more clients and didn't reduce the server performance; however, end users would experience more waiting time to get requests serviced. The second stage for 2 processors was from 28 clients to 112 clients.

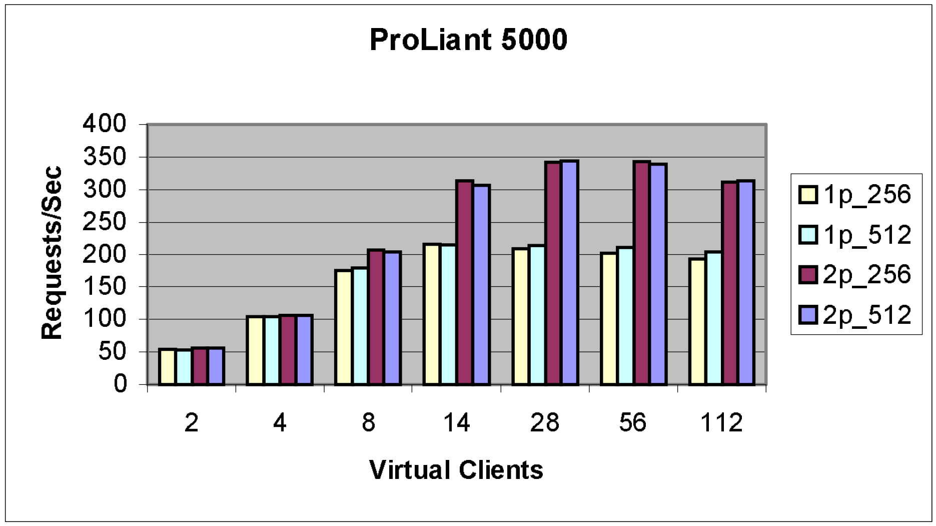


Figure 12: Throughput for ProLiant 5000 in Light CGI Model

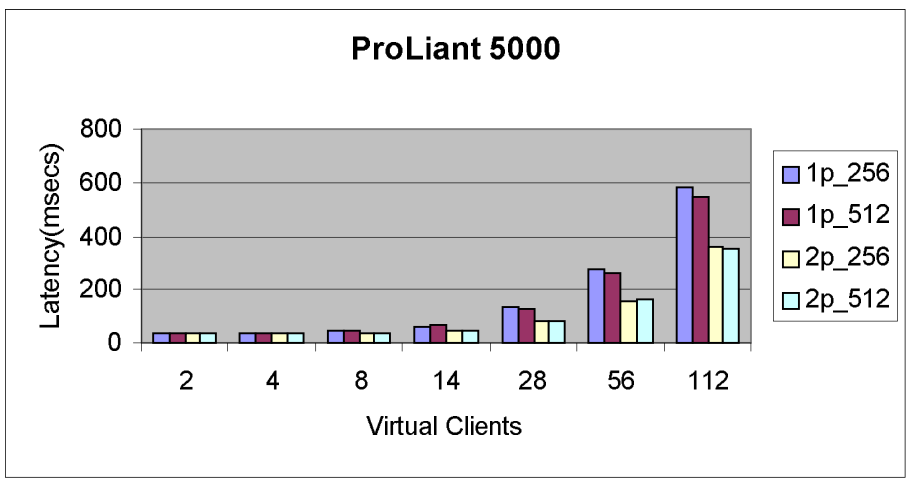


Figure 13: Latency for ProLiant 5000 in Light CGI Model

ProLiant 800

Figure 14 shows the benchmarking results on the ProLiant 800 with 1 Pentium Pro 200 processor and the ProLiant 800 with 2 processors. The 2-processor server shows approximately a 50% performance increase over the 1-processor server.

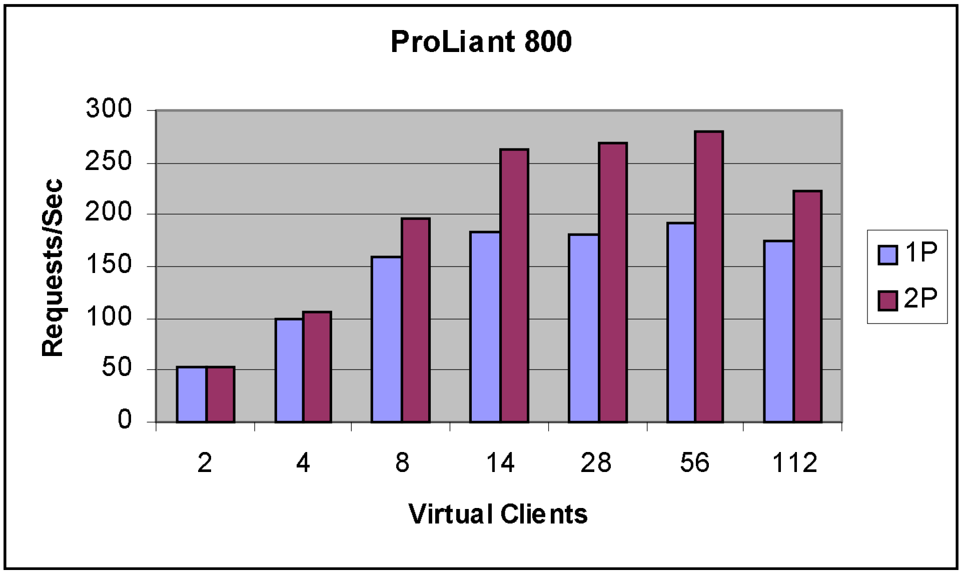
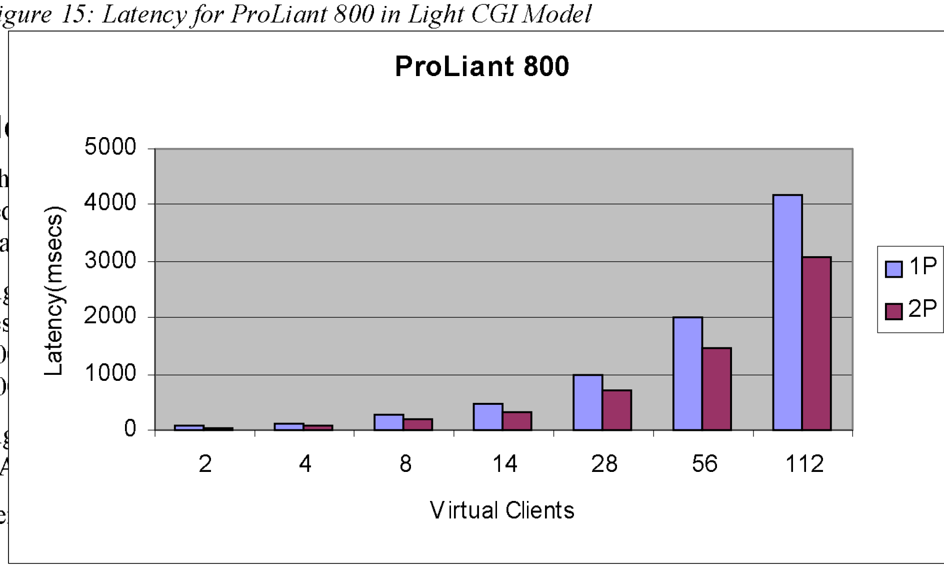


Figure 14: Throughput for ProLiant 800 in Light CGI Model

Figure 15: Latency for ProLiant 800 in Light CGI Model



H

Th … e CGI
rec … N card
wa

Fi … nt 800,
res … Liant
50 … ProLiant
50

Fi … 12 MB
RA

Pe … l.

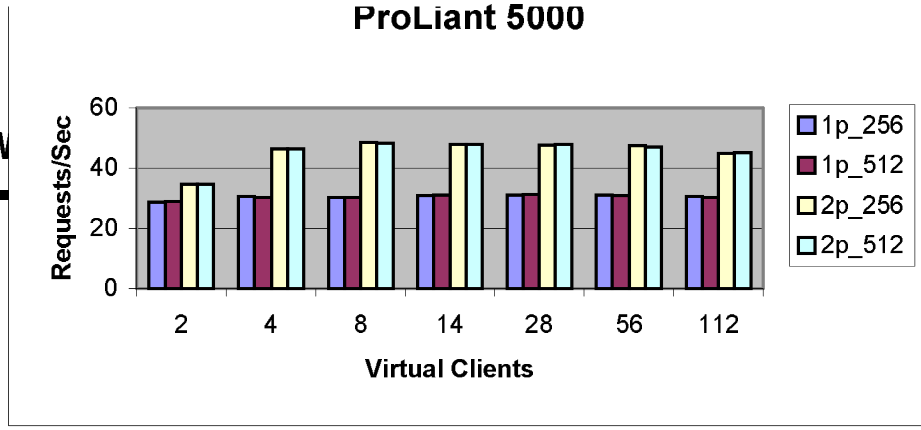
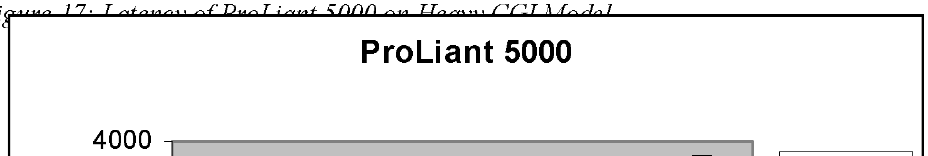


Figure 16: Throughput of ProLiant 5000 on Heavy CGI Model

Figure 17: Latency of ProLiant 5000 on Heavy CGI Model



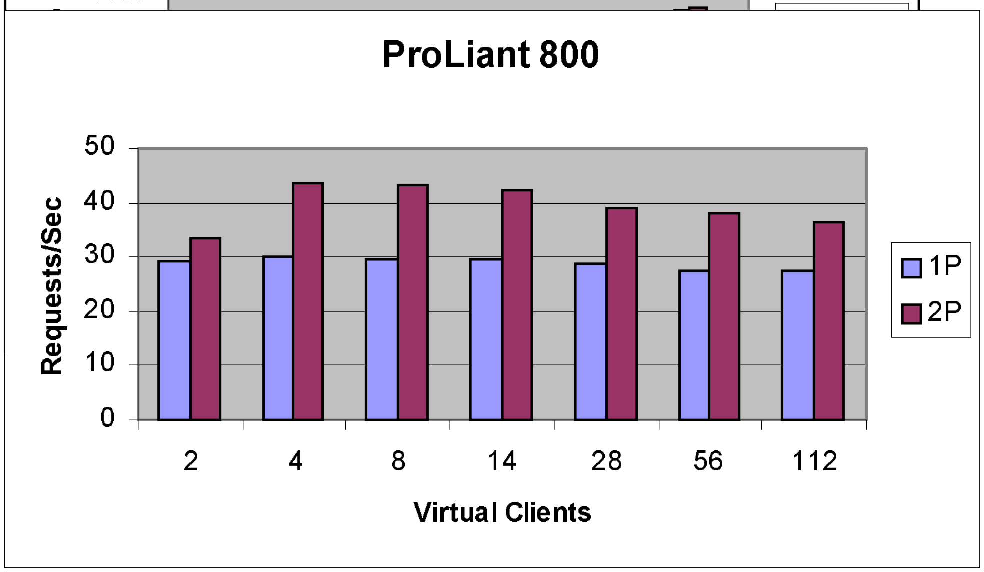
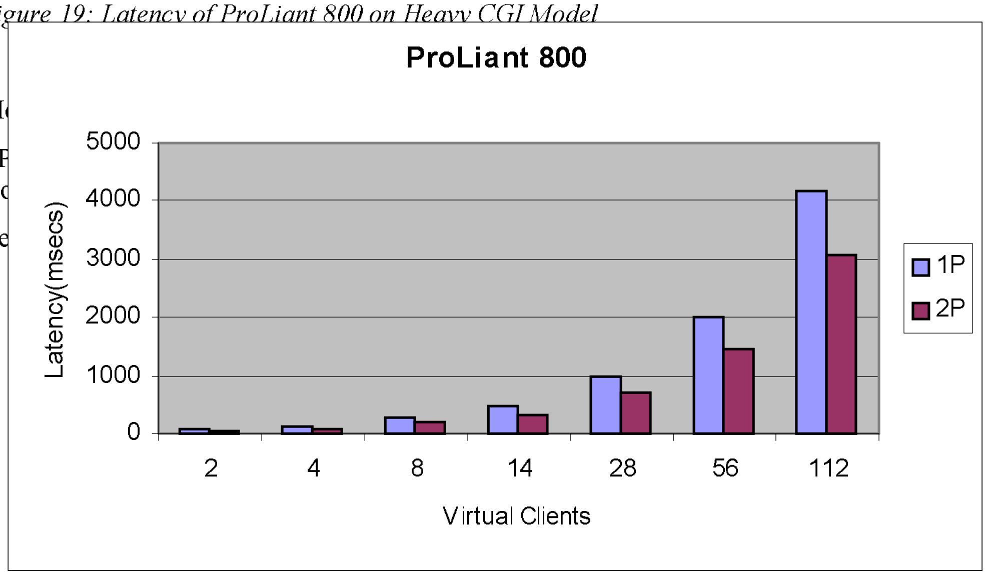


Figure 18: Throughput of ProLiant 800 on Heavy CGI Model

Figure 19: Latency of ProLiant 800 on Heavy CGI Model



SUMMARY

Peak Performance

Table 5 lists the peak throughput on benchmarking tests for the ProLiant 5000 and the ProLiant 800 servers in the three user-requirement models. The server throughput decreased significantly in the Light CGI and Heavy CGI models due to the CGI requests invoking a new server-side process. Even a small amount of CGI requests (as little as 10% in the Light CGI model), could cause a drop in server performance. When comparing the Light CGI model and the Static HTML model, the CPU utilization is almost the same in a peak-performance situation. The server on the Light CGI model can only generate about one quarter of the total throughput as the server working on the Static HTML model.

When the virtual client number initially increased for the three models, the throughput increased; however, when the throughput reached its largest output, increasing the virtual clients did not show a corresponding increase in server output. The server still maintains throughput near the peak number, although the latency increases simultaneously. Network collision rates are expected to increase when the client number increases. A decrease in server performance may be noticeable to end users.

Please note that WebBench uses virtual clients to simulate the Web browser users, but their behavior is very different. Unlike actual browsers, which take a great deal of time to display the files that the server sends, the virtual client keeps sending requests until it gets the reply from the server. The virtual clients are much busier than normal users. As in the benchmarking tests on the ProLiant 2P Static model, when there were 240 virtual clients, each client sent about 4200 requests during the 10 minute testing period. It is assumed the normal user will send much less than 4200 requests in 10 minutes.

TABLE 5. PEAK THROUGHPUT

Models:	Static HTML:	Light CGI:	Heavy CGI:
ProLiant 5000 1P	1072 Req/Sec	215 Req/Sec	31 Req/Sec
ProLiant 5000 2P	1401 Req/Sec	344 Req/Sec	48 Req/Sec
ProLiant 800 1P	742 Req/Sec	191 Req/Sec	30 Req/Sec
ProLiant 800 2P	1031 Req/Sec	279 Req/Sec	44 Req/Sec

Bottlenecks

Typical bottlenecks for a web server are:

- Networking
- CPUs
- Memory
- Disks
- Bus Architecture

Networking

For the Static model, networking is the most likely bottleneck. For benchmarking tests on the ProLiant 5000 2P, the networking utilization of two dedicated 100-Mbit Fast Ethernet LAN cards reached 50%-75%, which is probably the limit that Ethernet LAN can handle for the HTTP protocol. In this situation, a significant number of network collisions can be expected.

Since HTTP is probably not the only protocol present in an intranet, networking becoming a larger bottleneck to intranet performance is likely.

For the Light CGI model, networking would be the second bottleneck to performance after the CPU. However, in a 10-Mbit Ethernet LAN, the network will be the likely bottleneck. For the Heavy CGI model, networking does not seem to impair performance.

CPU

Adding a second CPU will always improve the server performance, limited only by network bandwidth. In the benchmarking tests, the performance of both the ProLiant 5000 and ProLiant 800 servers increased consistently from a 1-processor to 2-processor configuration for all three models. In a peak-performance situation, CPU utilization reached nearly 100% in the benchmarking tests, which is another indication that the CPU is the bottleneck.

For the Light CGI model and Heavy CGI model, the CPU will always be the bottleneck.

Memory and Disk

For all benchmarking tests, the memory and disk did not show any significant effects on server performance. For dynamic tests, the memory is probably not the bottleneck.

For the Static model, the results showed some inconsistency to other results released, most likely due to the small size of the sample files in the benchmarking tests. A large amount of memory was not required to cache all the files. For users with a large file set, memory could effect server performance more than what was shown in our tests.

Bus Architecture

Compaq also verified that a PCI bus performed much better than an EISA bus. The Compaq ProLiant 5000 and ProLiant 800 servers support PCI bus, so it is recommended that the customer select PCI cards, such as PCI NICs and PCI Array controllers, instead of EISA cards for use in these servers.

Capacity Solutions

For Internet/intranet users who are not satisfied with the current performance of their servers, or for users who are planning high capacity servers, the following suggestions will help in choosing efficient server platforms.

Requirements

Identify the load model

Based on performance research, it is clear that the servers perform differently in different user models. In a Static model, the ProLiant 5000 with 2P reaches about 1400 requests/sec, and the Light CGI model reaches only about 344 requests/sec, and as low as 48 requests/sec for the Heavy CGI model.

Understanding the load requirements is critical in identifying bottlenecks.

Identify the number of customers

In this testing, virtual clients were used to simulate real-world users. The virtual client number can not be directly converted to the number of real world customers. As previously stated, the behavior of the virtual client is different than that of the real-world user not only because the browsers take time to display the files they receive, but also because the customer normally spends time reading the files.

The tests on the ProLiant 5000 in the Static model with 120 virtual clients showed that the average latency is 92 milliseconds, and the throughput is 1330 requests/sec. In the 10-minute tests, the server handled about 798,000 requests, with an average latency of 92 milliseconds. Normally a browser takes only seconds to display information; however, it takes longer to display an image file. It is assumed that a user will have 5 requests/minute (which means that the user will generate about 2,400 requests in 8 business hours). The tested server can handle about 798,000 requests/5 requests per minute*10 minutes = near 16,000 users.

Efficiency Recommendations

Identifying the bottlenecks and analyzing the user load models is important. The following recommendations are based on the benchmarking performance tests:

- The network is the bottleneck for the Static model on a ProLiant 5000 or ProLiant 800 with only 1 dedicated 100-Mbit LAN.
- For a user with dynamic contents on their site, adding a second processor, and/or replacing a ProLiant 800 with a ProLiant 5000 greatly improves performance.
- A 10-Mbit LAN is a bottleneck for intranet users. Replace with a 100 Mbit LAN to improve the performance.
- Select a 2-processor server as the most cost-efficient solution for CGI models.
- Although performance was not sensitive to memory in the benchmarking tests, servers that handle a big file set will benefit from more memory.
- Select PCI bus cards for the NICs, array controller, etc.

APPENDIX A: DISTRIBUTION OF SAMPLE FILES

CLASS_256B	7
CLASS_512B	7
CLASS_1K	10
CLASS_2K	26
CLASS 4K	17
CLASS_8K	13
CLASS_16K	12
CLASS_32K	6
CLASS_64K	1
CLASS_128K	1

APPENDIX B: SAMPLE OUTPUT OF SIMCGI.EXE

SERVER_SOFTWARE = Netscape-Enterprise/2.01
SERVER_NAME = tjiang
GATEWAY_INTERFACE = CGI/1.1
SERVER_PROTOCOL = HTTP/1.0
SERVER_PORT = 8765
REQUEST_METHOD = GET
HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
HTTP_USER_AGENT = Mozilla/3.01Gold (WinNT; I)
HTTP_REFERER =
PATH_INFO =
PATH_TRANSLATED =
SCRIPT_NAME = /cgi-bin/simcgi.exe
QUERY_STRING =
REMOTE_HOST = 172.18.176.101
REMOTE ADDR = 172.18.176.101
REMOTE_USER =
AUTH_TYPE =
CONTENT_TYPE =
CONTENT_LENGTH =
ANNOTATION_SERVER =