WHITE PAPER

December 1997

Prepared By
Internet Solutions
Business Unit

Compaq Computer
Corporation

CONTENTS

Capacity Planning of Compaq ProLiant 850R with Netscape Enterprise Server 3.0

Capacity planning is a method of determining the balance between web server workload and its configuration, at minimum cost, while meeting necessary user response time objectives. The goal of capacity planning is to find the best server and equipment to cost-effectively meet network workload demands and performance requirements. Capacity planning allows for the balance of the supply and demand of present and future computer resources. Our key objective in capacity planning is consistent and acceptable user response times.

Compaq engineers have designed capacity tests and collected capacity data on Compaq ProLiant 850R with Netscape Enterprise Server 3.0, which are discussed in this whitepaper. A linear regression model was used to derive the formula based on the number of users and CPU usage. The capacity of each solution was defined as the number of users handled with 50% CPU usage. The capacity of ProLiant 850R was calculated as illustrated in the following table.

TABLE 1: PROLIANT 850R CAPACITY

Models	Solution	Capacity (Users)	Latency (Milliseconds)
Static	Solution 1	1100	10~14
	Solution 2	1800	12~18
Light CGI	Solution 1	375	17~20
	Solution 2	605	12~14
Heavy CGI	Solution 1	70	59~61
	Solution 2	100	53~70

The methods used to determine the options configuration, including memory, hard disks, and networking, are addressed in this paper. In addition, the three major procedures for successful capacity planning are given in the conclusion of this paper.

COMPAQ

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, SmartStart, and NetFlex, are registered with the 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, and Netscape Enterprise Server are trademarks and/or registered trademarks of Netscape Communication Corp.

Capacity Planning of Compaq ProLiant 850R with Netscape Enterprise Server 3.0

First Edition (December 1997)
ECG052/1197

DEFINITION OF CAPACITY PLANNING

Planning a hardware server platform that meets the needs of users, is one of the most important tasks of web administrators. Poor planning, whether over or underestimating computer resource needs, affects the corporate bottom line. When planners overestimate these needs, the result is a web server that costs too much, has excessive capacity, and is a waste of corporate resources. On the other hand, underestimating results in insufficient computer capacity. This can create unhappy users, affect group productivity, negatively impact a company's bottom line, and place the company at a competitive disadvantage. Planning for sufficient computer capacity is an ongoing process that allows for the avoidance of overspending and product insufficiency.

In the current drive to reduce corporate spending, the allowance for planning is sometimes trimmed, or even overlooked. However, reducing or neglecting this task exposes enterprise to insufficiently planned web server implementation and capacity. When this occurs, the web master/administrator spends a large amount of time, effort, and money reacting to user and management complaints and creating short-term fixes. Ideally, they should be providing support and participating in development and strategic planning functions, all of which are critical administrative responsibilities.

The amount of time, effort, and money spent properly planning web server implementation is worth the investment. The result is a system that adequately meets user response-time expectations and optimizes system resources.

This white paper provides information and methodology to help web masters and administrators effectively plan web server implementation.

Workloads

The most common question for capacity planners is, "How many users will the web server be able to support?". The most common answer is, "It depends", and as imprecise as it seems, this answer is the most accurate. The number of users handled by the same web server implementation can vary significantly for each different workload, making a single, precise answer impossible.

Generally the contents of a web site can be divided into two categories: static and dynamic. However, this paper presents three sample workload models that are illustrated in Table 2.

- Static HTML Model: In this model, regardless of whether the files are HTML or graphic, all the requests are static HTTP GET requests. Our analysis shows that more than 90% of sites belong to this model.
- Light CGI Model: In this model, about 10% of the requests are CGI, and 90% are static.
- Heavy CGI Model: In this model, about 90% of the requests are CGI, and 10% are static.

TABLE 2: MODEL OF WORKLOADS

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

Capacity Planning Metrics

The primary metrics of interest in capacity planning are utilization and response time, rather than throughput. In addition to these two metrics, the number of users, CPU usage, and latency, were also factors in our planning. In addition, the number of users at a particular CPU usage was a factor. The aforementioned metrics are briefly described below:

- *Utilization:* The amount of work being done by the server, as compared to the capacity of the server to do the work.
- *Response Time*: This is a number, usually measured in milliseconds or seconds, that relates the amount of time it takes a request to be completely processed and the results returned.
- *Latency*: The time taken by a client to retrieve a file from the server (measured in milliseconds).
- *CPU Usage:* Indicates how much of the CPU resource is used (measured by percentage).
- *Users:* We used one thread to simulate one user, so the number of total threads will equal the number of users.
- *Throughput*: This is the amount of data (measured in Mbits/Sec) that the server was able to process (measured in Mbits/Sec).

IDENTIFYING THE CAPACITY OF PROLIANT 850R

The capacity of the web server is defined as the number of users that the configuration can serve, with a different utilization level for workload models. 70% CPU utilization is generally considered the maximum capacity of a web server. That leaves 30% of the total CPU resources for other background tasks that the server might be running. Considering an additional 20% CPU resource for burst situation use, 50% CPU usage stands as a comfortable resource utilization point. The number of users that a web server configuration can handle with the CPU usage at 50% is defined as the capacity of that configuration. However, there are different capacities for each web server configuration, based on different workload models.

Compaq engineers designed the Web Server Capacity Simulation Test to identify the capacity of different web server configurations on three working load models; the Static HTML Model, the Light CGI Model, and the Heavy CGI Model.

Design of Capacity Tests

Capacity tests are different from performance tests in two aspects:

- Normally, in performance tests, the Server Under Test (SUT) will be stressed to nearly 100% CPU usage, and peak performance is the focus. In order to reach the peak performance, the resources are usually exhausted. However, in capacity tests, the focus shifts to the capability of the server at a particular resource utilization level.
- Although virtual clients can be varied in performance tests, the behavior of each client can not be controlled. It is very difficult to directly convert the number of virtual clients to real users without expertise in performance tests. In capacity tests, each client process (or thread) works as predefined, so the clients can be used to simulate real users.

There are several web server performance tools, such as WebBench, WebStone, and SPECweb96, but only WebBench has a client-control feature. With some configuration changes, WebBench was used as the load generator in our capacity tests.

Define User Behavior

Only one type of user was simulated in our capacity tests. This user spent 3 seconds between retrievals from the web server, and had 15-17 transactions per minute.

In WebBench 1.1, the thinking time of each client was set at 3 seconds to simulate our typical user.

Tests Environment Setup

Figure 1 shows an overview of the testing environment setup for the static suite test in the testing lab at Compaq. The controller was a ProLiant 4500 with one Pentium Pro processor and 256 MB of RAM. The controller sent the test information to the clients, and when the clients completed the tests, the results were sent back to the controller. The CPU usage of the SUT was collected from the controller as well.

The clients sent HTTP requests to the SUT and collected data. In static suite tests, 24 client machines were deployed in two dedicated 100 Mbit fast Ethernet LANs. One dedicated 100 Mbit LAN was used in the CGI tests, and the number of client machines was also reduced to 14, as 14 machines have adequate capacity for CGI tests.

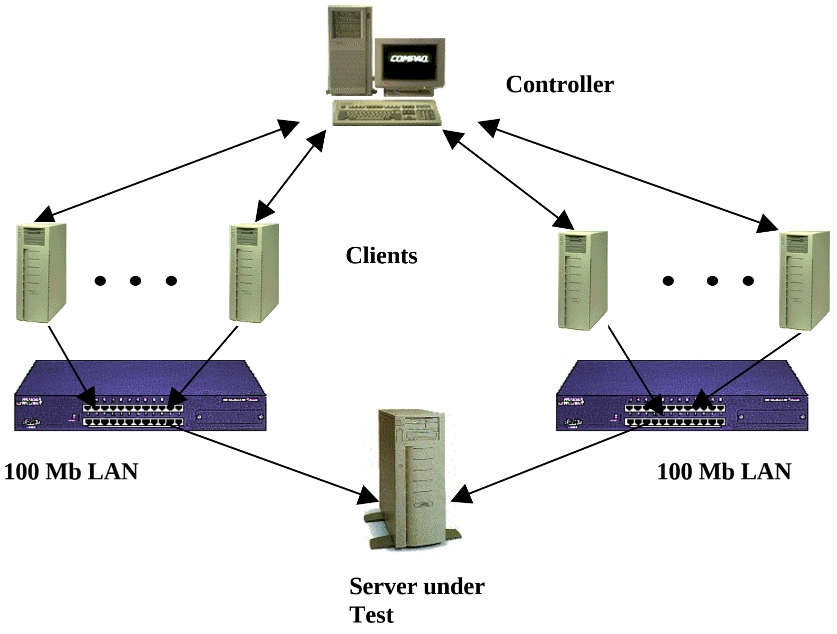

Figure 1: Overview of Testing Environment

There were 10 client processes running on client machines with Pentium Processors, and 20 client processes running on machines with Pentium Pro Processors. During the tests, the thread count of each client process was varied to simulate a different number of users. The behavior of each thread was the same, and is defined as follows:

- The thinking time was 3 seconds.
- The ramp-up time was 600 seconds.
- The length of each test was 1200 seconds.
- The keep-alive was turned off.
- The received buffer size of each client was 4K.

The ramp-up time was set at 600 seconds to limit the possible starting congestion due to the large number of clients. In addition, the length of each test was limited to 20 minutes to reduce the randomness of the tests. Although most current web browsers support keep-alive, they rarely have a chance to reuse the connections. In our tests, the keep-alive was turned off. Actually, most web server performance benchmark programs don’t support keep-alive.

Performance Monitoring and Latency

During the tests, PerfMon, running on the controller, was used to gather the CPU usage of the SUT. The CPU usage data were retrieved once every 15 seconds, and produced 40 effective data points for each test suite. The average CPU usage was used as the capacity metric. The average latency time of each request was calculated from the WebBench client reports, and was measured in milliseconds.

Configuration of the SUT

In our tests, we provided two web server solutions utilizing ProLiant 850R.

- Solution 1 used a Compaq ProLiant 850R with one Intel 200MHz Pentium Pro processor.
- Solution 2 used a Compaq ProLiant 850R with two Intel 200MHz Pentium Pro processors.

Other hardware configurations, identical for the two solutions, were as follows:

- 128 MB EDO RAM
- 2.1 Wide-Ultra SCSI
- Integrated 10/100 Ethernet Card

Both solutions had the same software configurations, as follows:

- Windows NT 4.0 Server with Service Pack 3 and updated Tcpip.sys
- Netscape Enterprise Server 3.0 with default configurations
- SmartStart 3.40, which can optimize the server configuration for Compaq ProLiant 850R, was used to install the software.
- Necessary management software, such as Compaq Windows NT Agents for Insight Manager, was running to ensure the SUT was as close to true running conditions as possible. In addition, the SNMP agent was turned on.

Capacity Metrics Observations

A sample set of data was selected to illustrate the relationship between the number of users, CPU usage, and latency. Figures 2 and 3 show the testing results of Solution 1, using the Light CGI Model. From Figure 3, it is clear that there are two stages of the test. In the first stage, as the number of users was increased, latency hovered around 20 milliseconds. However, when a certain number of clients was reached, latency increased dramatically. Latency was represented by a gently sloping curve, until the server began to be significantly stressed, then a very steep incline began. The change from the first to the second stage was defined as a transition point. In Figure 3, the transition point is at 800 users. Figure 2 shows the CPU usage in regard to the number of users in the same test. Again, the data clearly show two stages. From 100 to 700 users, as the number of users increased, the CPU usage increased linearly. After 700 users, the CPU usage reached 90%, and the SUT became stressed. Therefore, even if the number of users was increased, the server could not effectively support them. This is also evidenced by the increase of latency.

It can be concluded that, when the server is in an effective serving stage, there is a linear relationship between CPU usage and the number of users. As our tests could only be implemented discretely, we used a linear regression model to reflect the relationship between CPU usage and the number of users. Through the linear regression model, we can determine the capacity for our two web server solutions, based on different work models.

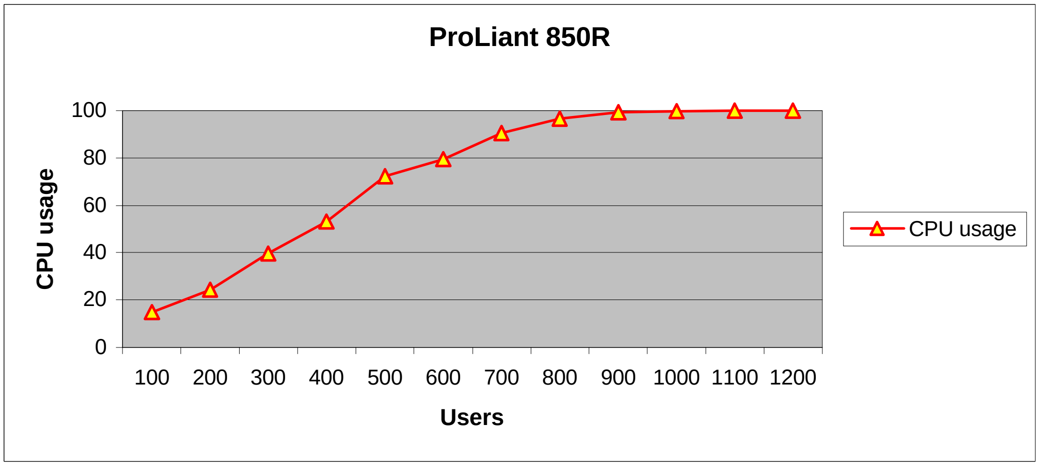

Figure 2: CPU Usage and the Number of Users (Light CGI Model)

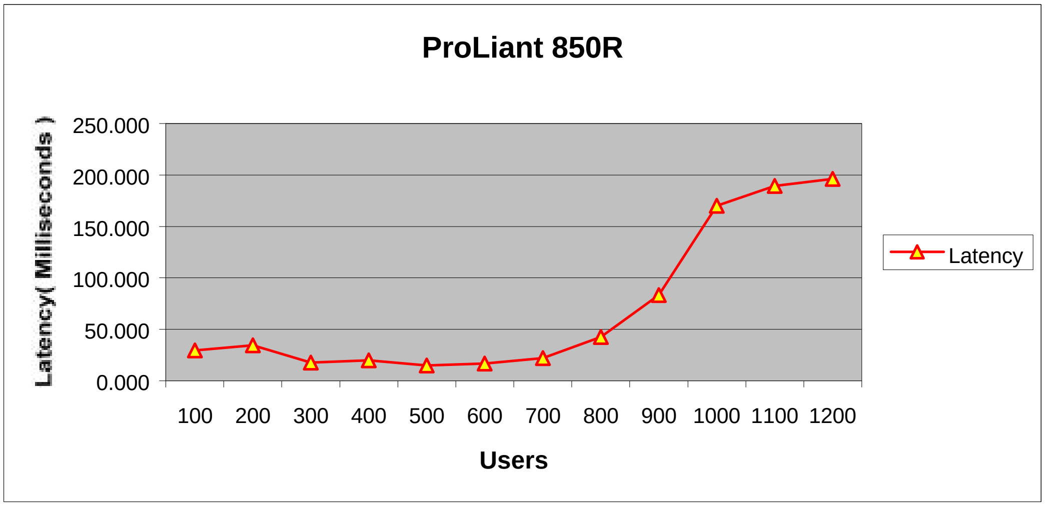

Figure 3: Latency and the Number of Users (Light CGI Model)

Methodologies of Linear Regression

The following is a formula, which represents the relationship between CPU usage and the number of users:

Formula (1) $Y=aX$

Y represents the CPU usage, and X represents the number of users. In order to determine the constant (a), a linear regression model was used. Only data before the transition point were considered valid. Table 3 shows the series of data for Solution 1 on the Light CGI model.

TABLE 3: LINEAR REGRESSION DATA

Number of users	CPU Usage
100	14.9
200	24.4
300	39.6
400	53.1
500	72.3
600	79.5
700	90.5

Figure 4 illustrates the linear regression results, for Solution 1, on the Light CGI model. The capacity formula is:

Formula (2) $Y=0.1334X$

The standard coefficient of determination is 99%, and the correlation coefficient is 99.7%. Both numbers indicated that CPU usage and the number of users were strongly correlated, and that the linear regression results were nearly perfect.

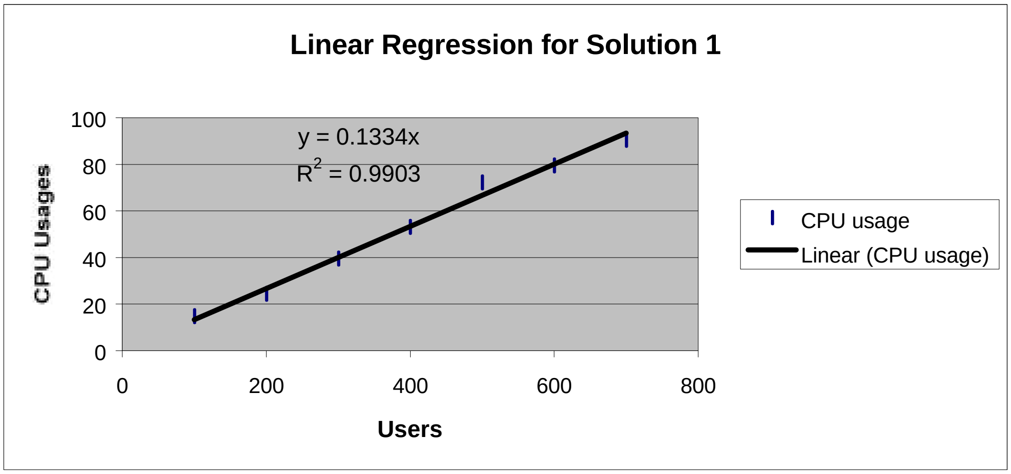

Figure 4: Linear Regression Results for Solution 1 on the Light CGI Model

As mentioned before, CPU usage for background tasks and burst situations was considered. Therefore, the number of users in this load model was set at 50% CPU usage capacity. Based on Formula (2), the capacity of Solution 1 on the Light CGI Model was calculated to be 375 users. Latency was selected from the two cases closest to this capacity. Latency at the capacity of 375 users was 17-22 milliseconds.

Results Analysis

The capacity tests were repeated for both solutions in all three workload models. The number of users was changed for different workload models, as the capacity of each solution varied. Tables 4, 5, and 6 show the results of the capacity tests. The first step in deriving the capacity formula is selecting valid data for the linear regression. As we know, the number of users and CPU usage showed a strong linear relationship when the server was in an effective serving stage. Only the data collected in this stage were used to do the linear regression calculations. In Tables 4-6, the *italicized* numbers were the transition points in each test, and only the data before this point were considered valid.

The transition points were selected with the following parameters in mind:

- There were latency jumps between test suites, or
- CPU usage increased linearly with the number of users.

In the Static Model test, the latencies of the first test suites were higher than the rest of the test suites. This was caused by congestion at the beginning of the benchmark program, which was evidenced by the connection time. The connection time is the time it takes the client to establish a connection to the server. This is almost as long as the transportation time, which is the time it takes the server to send a file to the client. Normally the connection time is only one-third of the transportation time. In the test group with 2560 users in the Static Model, the capacity limit of the benchmark programs and the testing client machine was reached, as evidenced by the 20 or so client processes that experienced system errors. Although Solution 2 was capable of handling more than 2560 users, latency increased substantially due to the capacity of the test bed.

TABLE 4: CAPACITY TESTING RESULTS (STATIC MODEL)

	Solution 1		Solution 2	
Users	Latency	CPU Usage	Latency	CPU Usage
320	97.446	13	42.870	9.078
640	19.297	29	18.715	15.285
960	13.991	44.8	12.874	26.647
1280	10.142	64	9.521	36.650
1600	11.469	70.4	7.873	42.784
1920	18.106	84.5	20.334	52.204
2240	*79.672*	*91.3*	28.134	62.923
2560	79.032	91.5	*42.202*	*67.211*

TABLE 5: CAPACITY TESTING RESULTS (LIGHT CGI MODEL)

	Solution 1		Solution 2	
Users	Latency	CPU Usage	Latency	CPU Usage
100	29.582	14.9	33.691	10.43
200	34.569	24.4	32.215	18.60
300	17.670	39.6	18.328	25.09
400	19.824	53.1	20.654	32.69
500	14.847	72.3	12.711	42.85
600	16.678	79.5	13.522	49.64
700	22.183	90.5	11.980	61.03
800	*42.619*	*96.7*	10.781	66.65
900	83.217	99.3	11.368	77.72
1000	170.219	99.8	15.576	81.71
1100	189.481	99.9	26.152	86.83
1200	196.096	100	*57.519*	*93.92*

TABLE 6: CAPACITY TESTING RESULTS (HEAVY CGI MODEL)

	Solution 1		Solution 2	
Users	Latency	CPU Usage	Latency	CPU usage
25	51.1	18.4	54.9	12.1
50	60.1	35.6	47.9	22.6
75	59.5	54.9	52.7	36.4
100	72.9	73.1	69.2	50.6
125	96.8	92.5	66.5	63.8
150	*391.7*	*98.5*	80.0	77.0
175	956.0	99.0	*410.8*	*83.1*
200	1664.0	98.0	651.8	88.7
225	1145.1	99.6	465.9	92.2
250	1423.3	100.0	566.2	96.6
275	1773.3	100.0	717.4	99.3
300	2014.6	100.0	938.9	99.9

In Table 7, which shows the linear regression results, the co-efficient of each formula is above 99%, and the coefficient of determination (R2) is above 98%. The linear relationship is strong and clear, and the results of the linear regression are quite positive.

Based on the capacity formula, the capacity of each solution was calculated at 50% of CPU usage, and latency was selected by choosing the two test groups that were closest to that number.

TABLE 7: LINEAR REGRESSION RESULTS

Models	Solution	Formula	Co-efficient	R2
Static	Solution 1	Y=0.0453X	99.30%	0.9861
	Solution 2	Y=0.0275X	99.70%	0.9951
Light CGI	Solution 1	Y=0.1334X	99.50%	0.9903
	Solution 2	Y=0.083X	99.70%	0.992
Heavy CGI	Solution 1	Y=0.734X	99.90%	0.9995
	Solution 2	Y=0.505X	99.90%	0.9963

TABLE 8: SOLUTION CAPACITIES AND POSSIBLE LATENCY

Models	Solution	Capacity	Latency
Static	Solution 1	1100	10~14
	Solution 2	1800	12~18
Light CGI	Solution 1	375	17~20
	Solution 2	605	12~14
Heavy CGI	Solution 1	70	59~61
	Solution 2	100	53~70

DETERMINING THE OPTIONS CONFIGURATION

In addition to the server capacity, there are three major components that also play important roles in capacity planning. These are:

- Memory,
- Hard Disk, and
- Network.

Memory Configuration

The Netscape Enterprise Server 3.0 boosts server performance, by using memory as its server cache, in order to avoid loading the static file from hard disk. As a result, the size of the required memory of the web server directly relates to the size of the static web content of the server.

When the size of the file set is small, such as 2MB, there are no significant effects on the memory configuration of the server. However, when the size of the file set increases, the effect on memory configuration becomes clear. Figure 5 shows the peak performance of a large sample file set at 50MB, with the memory configuration at 128MB RAM. This is almost 3 times more than the 64MB configuration.

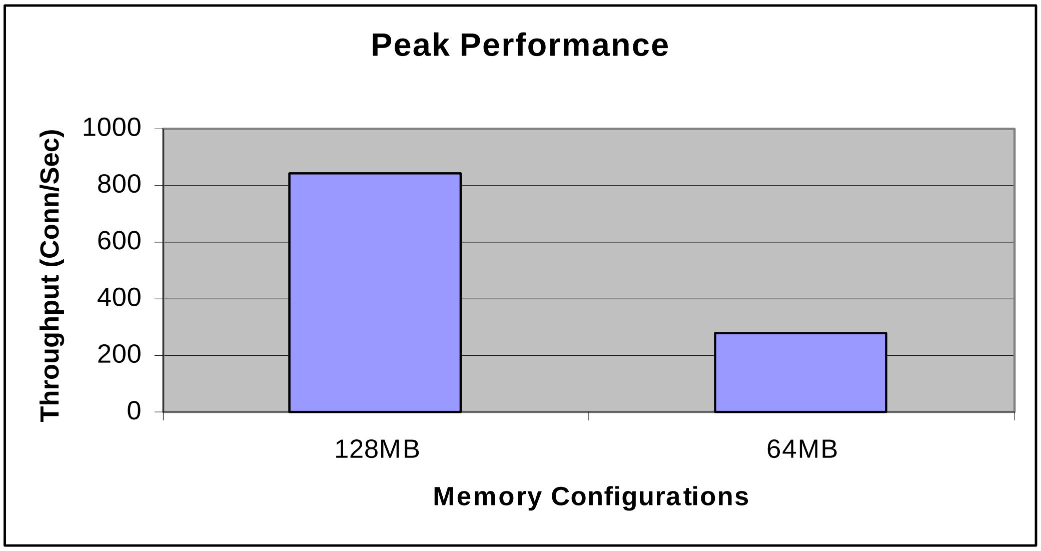

Figure 5: Peak Performance for Different Memory Configurations

In order to achieve the best throughput for web servers, memory should be big enough to cache all the static files. The NT 4.0 Server requires about 32 MB RAM, so the minimum memory requirement for peak performance is:

- Formula (3)   32+ the size of the static files

Our recommend memory configuration is:

- Formula (4)   32+1.5*the size of the static files

Table 9 shows our memory configuration recommendation.

TABLE 9: MEMORY CONFIGURATION RECOMMENDATION

Size of Static URLs	Minimum Memory Required(MB)	Recommend Memory (MB)
20 MB	64	64
50 MB	96	128
80 MB	128	196
120 MB	196	256
200 MB	256	384
300 MB	384	512

Hard Disk Configuration

The hard disk subsystem configuration of web server solutions should be set up with the following factors in mind:

- The disk space required by Windows NT 4.0, with Service Pack 3 near 200 MB.
- Paging space should be at least as large as the amount of RAM (twice the amount of RAM is recommended).
- Allow 100MB of disk space for the installation of the web server.
- Disk space for log files should be based on the total amount of access per day, with approximately 100 bytes allowed for each access.
- Double the size of the disk space for static web contents.
  - The minimum disk space required for our web server solution is:

  Total of hard disk space = 200+2*RAM+100+2*WebContents+0.0001*access

  Formula (5) = 400+2*RAM+2*WebContents (access set at 1 Million per day)
- The Smart-2 Array Controller is recommend for all servers that require fault-tolerance.

Network Configuration

Usually, the web server causes considerable network traffic. Our research shows that in the Static Model, the network is one of the possible web server bottlenecks. For end users, the network is the most likely bottleneck, as it generally has to share network bandwidth with other services. Fortunately, the traffic caused by the web server can be calculated, and we can approximately estimate the bandwidth consumed by the web server, allowing us to account for these problems in advance.

Table 10 gives the estimated traffic that would be caused by a different number of users on various workloads. The data in Table 10 were collected from the capacity tests.

TABLE 10: ESTIMATED NETWORK TRAFFIC (MBITS/SEC)

Heavy CGI Model		Light CGI Model		Static Model	
Users	Traffic	Users	Traffic	Users	Traffic
25	0.17	100	1.20	320	4.47
50	0.41	200	1.85	640	8.41
75	0.64	300	3.58	960	13.80
100	0.76	400	4.65	1280	22.43
125	1.00	500	7.08	1600	29.15
150	1.15	600	8.25	1920	34.86
		700	10.06	2240	39.72
		800	12.58		
		900	13.65		
		1000	18.36		

CAPACITY SOLUTIONS

Poor capacity planning, whether from over or underestimating the resources needed to set up an efficient server, will affect corporate interests. However, capacity planners can only plan for what is likely to happen; they have no way of knowing what will actually happen. There are simply too many possibilities to measure, and far too many unknowns. There is no fail-safe method of planning, but the information and methods provided in this paper offer an effective aid for plotting future hardware, software, and information systems strategies.

Capacity Solution Methodology

A summarization of the capacity tests done for ProLiant 850R and Netscape Enterprise Server 3.0, lead us to recommend the following capacity solution procedures for users.

- Clarify your objectives:

Before capacity planning begins, there are three questions that should be addressed. These questions are the foundation for good capacity planning.

- – How many users do you have?
- – What kind of workload model will fit your site?
- – What is the total file size of your static URLs?

The first two questions can be used to judge which server will fit your requirements, and the last question can help to determine the options configuration.

- Calculate the capacity:

Based on the answers to the first two questions, the projected CPU usage for each solution can be calculated by the capacity formulas given in Table 7. If CPU usage is below a certain level, such as 50%, the solution should be a capable one.

- Determine the options configuration:
  - – The memory configuration can be calculated using Formulas (3) and (4), or Table 9.
  - – The required disk space can be calculated using Formula (5).
  - – The projected network traffic can be found in Table 10.

Capacity Planning Case Study

A fictitious company, XYZ, wants to develop an Intranet server for 450 users. The web server is estimated to have a 10% dynamic content, and 90% static content. The size of the static content is estimated to be 50MB. The server will connect to a 100Mbits-shared hub, and the web server will use network bandwidth of about 20%. The following is a capacity plan for XYZ.

- Clarify the objectives.
  - – How many users will be supported?
  - – *Answer: 450 users.*
  - – Which loading model best fits the web site?
  - – *Answer: Light CGI*
  - – What is the total size of static URLs?
  - – *Answer: 50MB*

- Calculate the capacity.
  - For Light CGI model, the capacity formulas are:

  Y=0.1334X (Solution 1), and Y=0.083X (Solution 2).

Therefore, we estimate that CPU usage will be approximately 60% for Solution 1, and 40% for Solution 2. Since CPU usage for Solution 1 is over 50%, Solution 2 is more suitable for company XYZ.

- Determine the options configurations.
  - Memory:

    From Table 9, we know that the minimum memory is 96MB, and the recommended memory is 128MB. Thus, we would select 128MB of memory for company XYZ.

  - Hard Disk:

    Calculation is based on Formula (1), and the total amount of hard disk space=400+2*128+2*50=756 MB

    This is the basic requirement for the hard disk. If users have more information to be stored in the server, additional disk space may be required.

  - Network Bandwidth:

    Using Table 10, we know the possible network traffic generated by the web server will be between 4.65 Mbits/Sec and 7.08 Mbits/Sec.

    With a 100 Mbit shared hub, the possible network utilization for HTTP protocol is around 50-60 Mbits/Sec. The XYZ web server only uses 20% of the network, so the available network bandwidth for the XYZ web server is 10 Mbits/Sec. This is higher than the possible network traffic caused by the same server. Therefore, the capacity of the network is adequate.

Our capacity plan for the XYZ Intranet server is to use the ProLiant 850R, with 2 Pentium Pro processors with 128 MB RAM, a 2.1GB Wide SCSI hard disk, and 10/100 Integrated NIC. There will be no additional network segment, and the Netscape Enterprise Server 3.0 and Microsoft NT Server 4.0 will be installed.

Hints for Capacity Planners

The focus of capacity planning is to balance cost, and satisfactory user response time against a web server's workload. Effective capacity planning is interrelated with budgetary planning. The best solution is the one with the lowest cost and the best capacity, but it may not be easily achieved. It is important to try to balance the cost of a product with its effectiveness. Figure 6 shows the relationship between the TCO (Total Cost of Ownership) and the server's capacity. All of the solutions below the balancing line are good choices, with those furthest from the balance line being the best solutions.

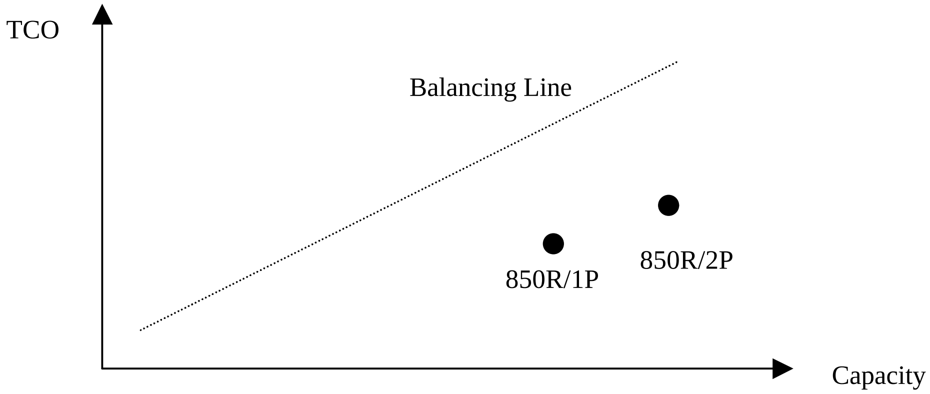

Figure 6: Capacity vs. TCO

In conclusion, we would like to offer some technical hints for capacity calculations.

- In our capacity tests, we only simulate one kind of typical user, with the user sending 15-17 requests per minute. If your user behavior is different, you can convert them to “typical” users. For example, if the users in your group normally send 5 requests per second, then you can divide the number of users by 3 to convert to the “typical” user.
- There are many different kinds of dynamic content. Some dynamic requests require additional resources, such as CPU time, memory, etc. Capacity planners have to make adjustments based on each individual case.
- Overestimating is bad, but underestimating is far worse.