Performance Analysis and Modeling of a WWW Internet Server

Virgílio A. F. Almeida*
Jussara Marques de Almeida
Cristina Duarte Murta†
Adriana Andrade Oliveira
Marco Aurelio de Souza Mendes

* Computer Science Department
Boston University
111 Cummington St,
Boston, MA 02215
virgilio@cs.bu.edu

Depto. de Ciencia da Computacão
Universidade Federal de Minas Gerais
Belo Horizonte, MG 30161
Brazil
{jussara,cristina,dri,corelio}@dcc.ufmg.br

Abstract

WWW users want fast and easy access to all documents available on the net. Thus, Web server performance is becoming increasingly important. The perceived Web latency i.e., the request response time, is affected by the performance of three components: the client, the server and the network that connects clients and servers. The client delay is the time required by the browser to show the document, considering the different types of media, such as audio, video, text, and graphics. The network latency represents the time required to provide the remote access plus the data transmission time. The server delay is the time required by the server to service the request. In this paper we focus on the analysis and modeling of the server performance.

1 Introduction

The World-Wide Web (WWW or Web) is a client-server system that integrates various types of information on the global Internet and on enterprise Internet Protocol (IP) networks [3]. It defines a global naming convention for all documents that are part of the Web. It allows users to navigate sites all around the world. The growth in popularity of the Web has been enormous. It is the fastest growing segment on the Internet. WWW technology involves the

*On sabbatical leave from Depto. de Ciência da Computacão da UFMG, Brazil. Partially supported by CNPq-Brazil.

†Depto de Informática da UFPR, Brazil. Partially supported by Capes-Brazil

combination of Web browsers and Web servers. The former provides an easy-to-use graphical interfaces for browsing information resources of the Internet.

WWW users want fast and easy access to all documents available on the net. Thus, Web server performance is becoming increasingly important. The perceived Web latency i.e., the request response time, is affected by the performance of three components: the client, the server and the network that connects clients and servers. The client delay is the time required by the browser to show the document, considering the different types of media, such as audio, video, text, and graphics. The network latency represents the time required to provide the remote access plus the data transmission time. The server delay is the time required by the server to service the request. In this paper we focus on the analysis and modeling of the server performance. The paper is organized as follows. Section two discusses performance of WWW servers. In section three, we give a general overview of a server environment. Section four presents experimental results that describe the server performance under different workload. In section five we show a simple queuing model of the server performance. Concluding remarks appear in Section 6.

2 Performance Measures of a WWW Server

A Web server is a system on a network that can process HTTP requests. The HyperText Transfer Protocol (HTTP) is the primary protocol used by the Web to retrieve information from distributed servers. Time and rate are the basic measures of a server's performance. The rate at which HTTP requests are serviced is known as throughput. Response Time is the time the server takes to process one request. In the study of performance of a Web server, we are interested in two measures: throughput and response time.

Benchmarking has been regarded as a useful approach for analyzing and predicting performance of computer systems. Several benchmarks have been proposed to measure hardware and software speed, including compilers and operating systems. The most often cited are SPEC, TPC, and Linpack [6]. WWW servers process specific workloads, composed mostly of HTTP requests. Thus, new benchmarks are needed to understand the server performance. WebStone is a configurable client-server benchmark for HTTP servers [7]. It uses workload parameters and clients to generate HTTP traffic that allows an HTTP server to be stressed in a number of different ways. This can give insight into the server behavior.

2.1 The WebStone Benchmark

WebStone makes a number of HTTP 1.0 GET requests for specific pages on a Web server and measures the performance of the server software and hardware platform set. WebStone is a distributed, multi-process benchmark [7]. The master process (WebMASTER) remotely spawns a number of clients to generate HTTP traffic based on the workload parameters specified in the configuration files. After all clients have been initialized, the benchmark is executed. As each of the clients finishes its run, the data are collected from each client by the WebMASTER and a performance report is generated.

The WebStone is designed to run for a specified duration. The maximum running time is dependent on the client machine memory and the number of client processes to be spawn. It also has the ability to run for a number of iterations. The number of client machines and client processes per machine are configurable. The clients and the WebMASTER may run

on the same machine or not. The number of clients processes per machine is limited only by the machine memory. Load generation in WebStone is done by successively requesting pages and files from the server as fast as it can send them, reflecting the current environment in the WWW community. The load generation is based on two configuration files: testbed and filelist. The former specifies a lot of information including the number and name of the client machines, the number of iterations, the test time and the number of client processes. The latter specifies the number of pages to be requested and the type of these pages. Each page is a set of HTML files of different sizes. The type of a page is mainly determined by its size (number of files) and its access probability. A request for a page represents a request for each one of its files.

The WebStone primary results are throughput measured in Bytes/second and latency, which is the average response time to complete a request. Other measures are connection rate averages and Little's Load Factor, derived from Little's law [6]. The last one reflects how much time is spent by the server on request processing, not including errors and overhead. It is a measure of the effective utilization of the server. Ideally, Little's Load Factor should always be equal to the number of the client processes. A lower Little's Load Factor indicates that the load on the server is high, and some clients are not being serviced before they time out.

3 The Server Environment

The experiments were carried out in a dedicated server platform, with the following characteristics: Pentium 100 Mhz processor, 16 Mbytes of memory, Windows NT Workstation 3.5 with TCP/IP and Ethernet network connection. The Web server software is the EMWAC HTTPs, that originates from the European Microsoft Windows NT Academic Centre [5]. It is designed for computers running the Windows NT operating system and it implements the HTTP/1.0 protocol. It runs as a "service" and is called *https*. The server platform was connected by a non-dedicated Ethernet network to a Unix machine where the WebStone runs. WebMASTER and all clients processes were started from the same machine, an SparcStation with 256 Mbytes of RAM running SunOS 5.4.

The performance of a Web server depends on its hardware and software. From the hardware standpoint, the main components are: processor, memory, disk and network. From a software standpoint, the most relevant components to the server's performance are the HTTP service, the TCP/IP implementation, and the operating system, which is responsible for the file subsystem.

3.1 Windows NT and Performance Monitor

Windows NT is an operating system that supports multitasking and multithreading, allowing users to run multiple programs at the same time. A NT process consists of an executable program, a set of virtual memory addresses and at least one thread, which is an executable entity that belongs to one process [3]. Two or more threads can be created to execute within a single process, sharing the same address space and other resources, improving the performance. Multithreading processes have been used as an ideal solution to implement server application in a client/server environment. NT has been widely used as operating system of various types of servers such as: database, application, network, and WWW.

Object	Counter	Description
Memory	Pages/s	Number of pages transfered from and to disk
Physical Disk	% Disk Time	Percentage of elapsed time the disk was busy
Processor	% Processor Time	Percentage of elapsed time the CPU was busy
	Interrupts/sec.	Number of device interrupts
Processes	% Processor Time	Percentage of CPU time used by http
http	Thread Count	Number of active http threads

Table 1: Description of the counters and objects monitored

Windows NT provides a set of performance tools that can be used to collect and display performance information of any computer in a distributed environment. Performance Monitor [1] is a graphical tool based on a series of counters that represent performance measures. It allows a user to study the behavior of resources such as processors, memory, logical disks, processes and network interface and identifies the resources as objects. A unique set of counters exists for each object. The concept of a counter is fundamental to understand NT performance. A counter is an entity for which performance data is available. They represent measures such as processor and disk utilization, number of network packets transmitted per second, number of interrupts per second, number of threads of a specific processes, etc.

4 Experimental Results

A series of experiments were carried out to investigate performance of a WWW server. The experiments consisted of the execution of the WebStone processes in the Unix machine with the workload described by the parameters of the filelists. The experiments were monitored using the logging capability of Windows NT. The log files were retrieved back by Performance Monitor for charting. The objects monitored were memory, physical disk, processor and process. The description of each counter of the objects is shown in Table 1.

The experiments were done using two different workloads (A and B), whose main characteristics are shown in Table. 2 The file sizes of workload B are equally distributed between 1 and 200 KBytes. In the case of workload A, 94% of accessed files are less than 50 Kbytes. Figure 1 shows the results obtained by Webstone and Performance Monitor for the two workloads. At the client side, the Webstone processes collected the connection rate and the throughput, which are shown in the left-hand side graphic of figure 1. At the server side, Performance Monitor measured the processor utilization as a function of the number of clients. It also shows the processor utilization accounted for the HTTP service.

Figure 1 shows that under 20 clients, the connection rate grows as we increase the number of clients. Over 20 clients, the connection rate decreases due to the CPU load that reaches almost 100%. When the CPU can not handle more requests, i.e., gets saturated, the connection rate reaches its maximum value. The throughput and connection rate curves are similar in terms of shape. In our experiments connection rate and throughput have the same meaning. The difference is that connection rate is measured in HTTP requests/sec. while throughput is specified in bytes/sec. As the number of clients increases, the number of TCP/IP packets arriving at the server also increases. Since the CPU has to handle all

Features	Workload A	Workload B
Number of Files	18	189
Total size of the file system	298 Kbytes	5 Mbytes
Average file size	15 Kbytes	27 Kbytes

Table 2: Characteristics of the Workload

interrupts, the CPU usage by the HTTP processes decreases and, as a consequence, the connection rate. Since the average file size is the same, the throughput also decreases. Another important issue in performance of a Web server is the average response time. In

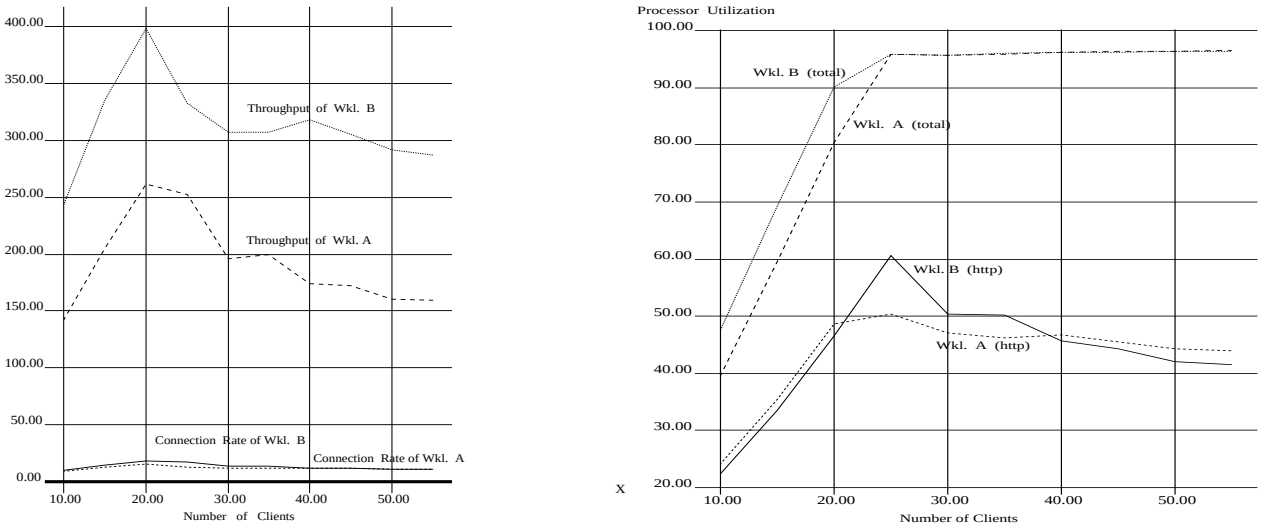

Figure 1: Client-server measures: connection rate (conn/s), throughput (KBytes/s) and *https* and total processor utilization (%) for the two filelists.

figure 2, we show the response time curve as a function of the number of clients and the interruption rate measured at the server, for the two workloads. The response time curve, as expected, increases with the number of clients. We can observe that the two curves of the right-hand side graph have the same behavior. They grow up until 40 clients and then they seem to get stable. We believe that this stability is due to the overloading of the CPU and the network interface. Some of the TCP/IP packets are lost and the corresponding interruptions are not created. Observe that the two curves cross around 40 clients. This observation can also be noted in the Response Time and Processor Time curves. After the crossing-point, the interruption rate for workload A became higher than that for workload B and the corresponding curve for the HTTP processor time became lower than the other one. This observation stems from the fact that the CPU has to handle more interruptions. As another consequence, the response time for workload A is higher than for workload B. Figure 3 shows the behavior of disk and memory. The curves of pages/s and disk time utilization are similar. Neither the memory nor the disk are bottlenecks. We must notice that the increase in the disk file utilization for the second workload is due to number of

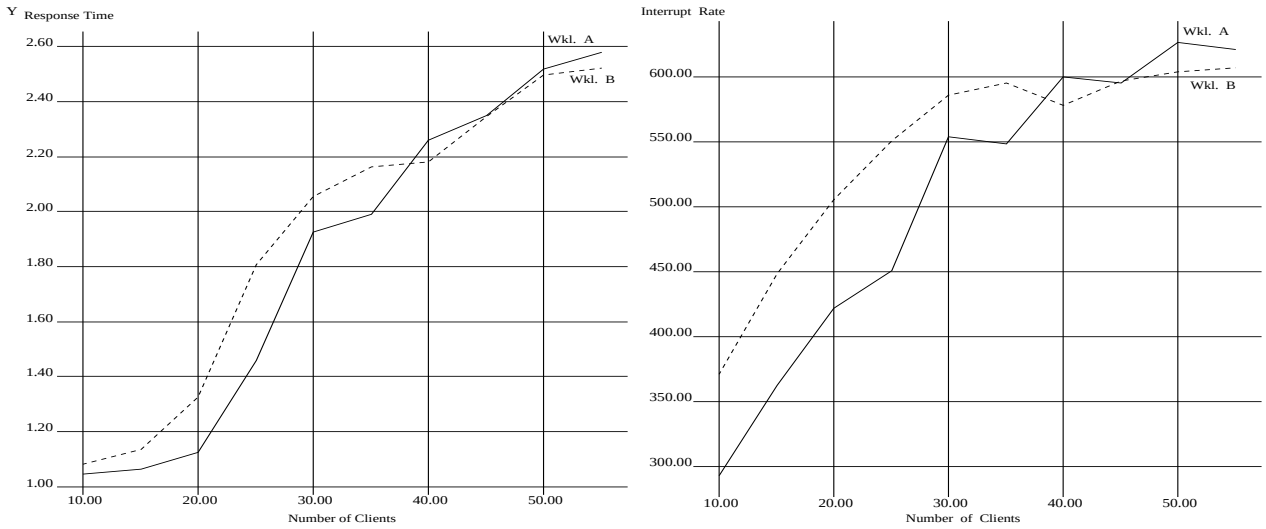

Figure 2: Client-server measures: response time (seconds) and interrupt rate for the two filelists.

different files and the total size of them. In the first configuration, all the files can be held in the cache at the same time, so that the disk accesses are serviced by data held in the cache. This situation does not repeat for the second workload.

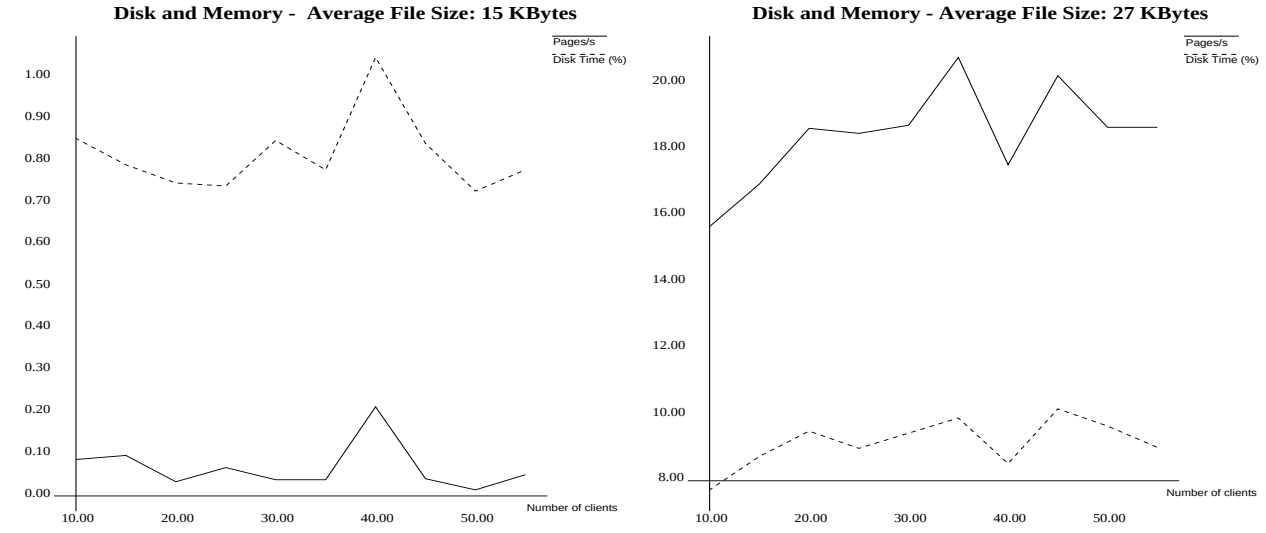

Figure 3: Performance Monitor's measures: Disk and Memory utilization

5 Analytical Model of a WWW Server

Our initial approach to server modeling is to develop a simple queuing network model representing the server architecture (processor, memory, and disks). The workload consist of one class: the HTTP requests that arrive at the server. The operating system overheads will be implicitly represented in that class.

The server is represented by a closed model, with the multiprogramming level equals to the number of connections in execution. A single-class closed model can be easily solved by the MVA (mean value analysis) equations [6]. The basic input parameters for a queuing model are the service demands, which represent the average total service time that an HTTP request spends at the processor and disks. The service demand at device i is given by the following expression from reference [6]:

$$D_i = T/C \times U_{global} \tag{1}$$

where D_i is the processor demand, T is the observation period, C is the total number of HTTP requests completed in T (given by the Webstone statistics), and U_{global} is the processor utilization registered by the counters of Performance Monitor. Figure 4 exhibits a graph of the connection rate of workload B measured in the experiments and the connection rate calculated by the queuing model. The quality of the prediction is reasonably good, especially when one looks at the second part of the results, that correspond to the values after the throughput reached its maximum value. In the first part of the curve, the model did not capture the behavior of the WWW server. The reason is that under 20 clients the number of requests is lower than the maximum the server can handle. The measured throughput reflects the situation, but the closed model assumes that the current number of clients is the maximum, so the predicted throughput is higher.

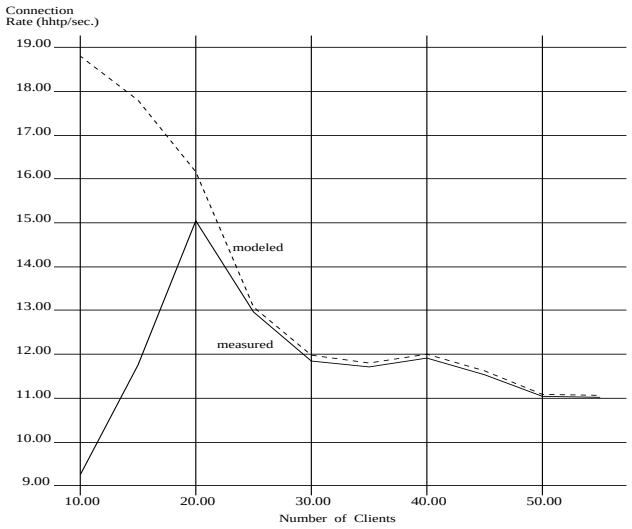

Figure 4: Server Performance: predicted and measured

6 Conclusions

This paper presents a performance analysis and modeling study of a WWW server. Using standard performance tools provided by the Windows NT operating system and the Webstone benchmark, we carried out a series of experiments to monitor the behavior of a WWW server. We using the two tools, it was possible to investigate the bottlenecks of a server. It was also possible to look at the influence of the operating system (e.g., cached file system)

on the performance of a WWW server. With the standard tools provided by the operating systems and Webserver softwares, it is difficult to obtain the input parameters required by queuing network models. We also presented a simple queuing model that reasonably represents the behavior of a saturated Web server.

References

[1] V. Almeida and G. Fialho, "Performance Analysis and Modeling of a Windows NT Server", Proceedings of CMG95, Nashville, December 1995.

[2] M. F. Arlitt and C. L. Williamson, Web Server Workload Characterization: The Search for Invariants, Department of Computer Science, University of Saskatchewan, Canada, October 1995

[3] T. Berners-Lee, R. Cailliau, A. Luotoneu, H. Nielsen, and A. Secret, "The World Wide Web", *Communications of the ACM*, Vol. 37, No. 8, August 1994.

[4] J. B. Chen, Y. Endo, K. Chan, D. Mazières, A. Dias, M. Seltzer and M. D. Smith, "The Measured Performance of Personal Computer Operating Systems", Proceedings of the 15th ACM Symposium on Operating System Principles, 1995

[5] European Microsoft Windows NT Academic Centre, HTTP Server Manual Version 0.99.

[6] D. Menascé, V. Almeida and L. Dowdy, *Capacity Planning and Performance Modeling: from mainframes to client/server systems*, PTR Prentice-Hall, Englewood Cliffs, 1994

[7] G. Trent and M. Sake, "WebSTONE: The First Generation in HTTP Server Benchmarking", MTS Silicon Graphics, February 1995