Appeared in the International Journal of High Performance Computing Applications, Vol. 14, No. 4, Winter 2000.

Web Performance Modeling Issues

Daniel A. Menascé
Department of Computer Science, MS 4A5
George Mason University
Fairfax, VA 22030-4444
USA
http://www.cs.gmu.edu/faculty/menasce.html

Abstract

The web has experienced phenomenal growth and has become an important element of the information infrastructure of many organizations. Web performance and around-the-clock availability are the major concerns of Webmasters of sites that support mission critical applications, such as electronic commerce. Web sites are complex computer systems consisting of many interconnected computers, networks, and routers. Some of the computers have specific functions such as certification or caching. Workloads on Web sites are often unpredictable in nature. Heavy load spikes and a large variability of the objects requested over the Web are unique characteristics of Web workloads. This paper reviews the important issues and challenges in modeling Web-based systems. Workload characterization, predictive models, and their use in various situations are discussed.

1. Introduction

According to a report entitled The Emerging Digital Economy (*US Department of Commerce, 1998*), "… IT and electronic commerce can be expected to drive economic growth for many years to come." In fact, a recent forecast shows that by the year 2,001, over $205 billion will be transacted over the Web, up from $11 billion in 1997 [9]. Being able to sell directly to consumers, bypassing the broker or agent, has proven to be successful and cost-effective in many industries. Some notable examples include the sales of computers (e.g., Dell Computers), airline tickets (e.g., Travelocity) or stocks (e.g., T. Rowe Price).

The World Wide Web has had a tremendous impact in most human activities. In its early days, it was mainly used to locate and retrieve documents from the Internet. With the advent of CGI (Common Gateway Interface) scripts, the WWW started to be used to access remote databases. Many traditional client/server applications were then replaced by Web-based ones running on companies' intranets. Eighty five percent of America's Fortune 100 companies have a presence on the Web and spent between $0.75 and $1.25 million on their web site. The next obvious step for the WWW was to become a vehicle for doing business. Electronic commerce (e-commerce) has been growing in importance at a very fast pace.

Many recognize that a major impediment for the growth of e-commerce is congestion. Congestion can occur in many places including LANs, WANs, routers, and servers. In the Web, the competition is a click of the mouse away. Therefore, it is imperative that e-commerce sites have acceptable performance or customers will quickly switch to other sites.

Section 2 of this paper provides some interesting statistics on e-commerce. Section 3 reviews some of the important facts about the characteristics of WWW traffic. The next section discusses how caches and prefetching at various levels can be used to improve performance. Section five discusses performance prediction for the Web showing how the effects on performance of phenomena such as burstiness and heavy tailed distributions can be taken into account by performance models. Section six presents a numerical example of performance prediction of an intranet with a proxy cache server. Finally, section 7 presents some concluding remarks.

2. Electronic Commerce Facts

More and more companies are moving into offering their products on the Web because it cuts costs and increases sales at the same time. By the end of February 1998, 10% of Dell computer sales took place through their Web site at

greatly reduced costs [17]. Mutual fund investors are finding it convenient to review their accounts on the Internet before making investment decisions and because they can make use of many on-line resources (e.g., stock market analyses, calculators, and fund results) available at the Web sites of mutual fund companies [25]. Fifty five thousand of T. Rowe Price investors are now using its Internet based trading service [25].

Some other interesting numbers about e-commerce are shown below [9]:

- Businesses will exchange $327 billion in goods and services by the year 2,002.
- Cisco Systems sells $4 billion a year on the Web at a cost savings of $363 million.
- General Electric estimates that e-commerce will save the company $500 million over the next three years.
- Boeing booked $100 million in spare parts in the first seven month of activity of its Web site.
- Texas Instruments fills 60,000 orders a month through its Web site meeting delivery deadlines 95% of the time.

A breakdown by sector of 1997's e-commerce activity and a forecast for the year 2,001 is given in Table 1. Event ticket sales is the sector with the largest predicted growth (2531%) followed by business-to-business transactions (2287%).

Table 1 - E-commerce in 1997 and forecast for 2001.

Type of Business	1997 in $billion	2001 (forecast) in $billion
Business to Business	8.000	183.000
Travel	0.654	7.400
Financial Services	1.200	5.000
PC Hardware & Software	0.863	3.800
Entertainment	0.298	2.700
Event Ticket Sales	0.079	2.000
Books & Music	0.156	1.100
Apparel & Footware	0.092	0.514
Total	**11.342**	**205.514**

GiveN the increasing wide spread use of the Web to support mission critical applications, such as e-commerce and others, it becomes very important to be able to use capacity planning techniques to these new environments. A methodology and models for capacity planning of Web environments is presented in [19]. The two most important steps of this methodology are workload characterization and performance modeling and prediction. Next section describes the main characteristics of WWW workloads as observed by many different studies.

3. WWW Traffic Characteristics

Web traffic is bursty and unpredictable in nature and has been shown to be self-similar, i.e., it is bursty over several time scales [5, 12]. Load spikes can be many times higher than average traffic. For example, consider a Web site that sells tickets for rock concerts. The load at this site is expected to surge considerably when Madonna announces a performance. As another example, consider what happens to a sport events site during big events such as the Olympic games or the World Soccer Cup.

Several workload characterization studies for the Web have been performed at various levels: client, proxy cache, server, and the Web in general. A list of most of the existing studies can be found at [23]. We review here some of the results found in workload characterization studies in each of these four categories.

3.1 Workload Characterization of Client Requests

In [14], the authors report on a detailed study of data collected from an instrumented version of NCSA's Mosaic. The log contained over half a million requests obtained in an academic setting. The authors showed that the distribution of document sizes, popularity of documents as a function of size, distribution of user requests for documents, and number of references to documents as a function of their overall rank in popularity can be modeled by power-law distributions. Some of the results found in [14] are:

- 22% of the requests generated by the browser were cache misses.
- 96% of the total requests were for html files and only 1% for CGI bin requests. With the advent of more search engines and electronic commerce sites, the percentage of dynamically generated pages increased since the study in [14] was conducted. More recent studies report a percentage of dynamically generated pages ranging from 2 to 6% [21].
- 78.6% of requests were for external servers
- Less than 10% of requests were for unique URLs, i.e., URLs not previously referenced.
- 9.61% of accesses were to html files with an average size of 6.4 KB and 69% to images with an average size of 14KB.

In [24], the authors report on the revisitation patterns in WWW navigation. Their study covered six weeks of WWW usage from 23 users. They concluded that 58% of the pages visited are revisits. Their study also shows that users tend to visit pages just visited more often than pages visited less recently.

3.2 Workload Characterization at the Proxy Server

A proxy cache server acts as a server for a Web browser. If the requested document is not in the cache of the proxy server, it acts as a client to the remote server. When the document arrives, it is sent to the client browser and stored in the cache of the proxy server. Proxy caches can reduce the latency perceived by Web users and reduce the bandwidth requirements at the link that connects an organization to its Internet Service Provider (ISP).

In [1], Web traffic from three different educational sites was collected during one semester and used to drive a trace-driven simulation of a cache proxy server. The authors found that the maximum cache hit rate was between 30 and 50% for infinite size caches regardless of cache design.

3.3 Workload Characterization of Web Server

.

Arlitt and Williamson [5] conducted an extensive study involving six World Wide Web servers including both academic and commercial sites. The number of requests analyzed for these sites ranged from 188 thousand to close to 3.5 million requests per site. The purpose of their study was to look for *invariants* in the characteristics of the workload of Web servers. We list here some of the important invariants found in their study:

- HTML and image files account for 90-100% of requests. This is consistent with the results reported in [14] from the client side.
- The average size of a transferred document does not exceed 21KB. Again this is consistent with results found in [14].
- Less than 3% of the requests are for distinct files.
- The file size distribution is Pareto with $0.40 < \alpha < 0.63$. In other words, this distribution is heavy-tailed and its probability mass function is given by

$$p(x) = \alpha \quad \frac{k^{\alpha}}{x^{\alpha+1}} \qquad \alpha, k > 0, x \geq k$$

- Ten percent of the files accessed account for 90% of server requests and 90% of the bytes transferred.
- File inter-reference times are exponentially distributed and independent.
- At least 70% of the requests come from remote sites. These requests account for at least 60% of the bytes transferred. This is consistent with the results in [14] that show that 78.6% of client requests went to external servers and the remaining to the local Web site.

Crovella and Bestravos [12] studied traces of users using NCSA's Mosaic Web browser reflecting requests to over half a million documents. The purpose of the study was to show the presence of self-similarity in Web traffic and attempt to explain the phenomenon through the understanding of various underlying aspects of the WWW workload such as distributions of WWW document sizes, the effects of caching and user preferences in file transfer. Similarly to [5], [12] reports that file sizes have a heavy-tailed distribution. This study also indicates that this distribution may explain the fact that transmission time distributions are also heavy-tailed.

Almeida and Oliveira [4] used fractal models to study the document reference pattern at Web servers. Fractals are self-similar mathematical structures characterized by distributions of the form

$$P(U > u) = (u / u_0)^{-\theta}$$

where u_O is a constant and θ is called the fractal dimension.

In [4], an LRU stack model [13] is used to study references to documents stored in a Web server. Trace logs from two different sites were analyzed and the reference strings and corresponding distance strings (i.e., distance of the referenced document from the top of the LRU stack [13]) were computed. Time was divided into bins of duration δ time units. The sum of the distance values in each bin was plotted versus time. The shape of this plot exhibits distance peaks and looks pretty much the same even when the value of δ is scaled over three orders of magnitude. In other words, there is strong evidence of self-similarity in the document reference pattern.

Another study on the reference locality in the Web was conducted by Almeida et al. [3].

3.4 Web Traffic Workload Characterization

In [8], a study was carried out to answer some quantitative questions about the Web in general. This study, performed in 1995, analyzed over 11 million Web pages. Some of the values found by this study are:

- The average page size was 6,518 bytes with a standard deviation of 31,678 bytes.
- About 50% of the pages were found to have at least one embedded image and 15% were found to have exactly one image.

- Over 80% of the sites are pointed by a few (between 1 and 10) other sites.
- Almost 80% of the sites contain no links to off-site URLs.
- Around 45% of the files had no extension and 37% were html files. Then .gif and .txt files were the next most popular with 2.5% each.

Next section describes some techniques that can be used to improve the performance perceived by Web users.

4. Improving Web Performance

Of the many techniques aimed at reducing latency, caching is one of the most cost-effective. Caching entails keeping a copy of a recently accessed object at a location (the cache) from where it can be retrieved faster than from its original site (the originating server's disk). Caches are commonly found at the browser and at proxy servers [6] located within the user's intranet.

Caches can also be used in a pre-fetching mode, i.e., if there is a way to predict what documents the user will request in the future, these documents can be fetched from the originating server ahead of time so that they will be already in the cache when requested.

4.1 Prefetching Inlines

Dodge and Menascé studied the use of pre-fetching inlines from HTML documents into the server's main memory cache, reducing disk access time when these, typically large files, are automatically requested by the web browser [15]. They suggest the scheme shown in figure 1 in which the server also parses the request while the HTML document is being sent to the browser and prefetches the inlines from disk into its main memory cache. This way, when the requests for inlines arrive at the server, they can be served from the cache immediately without having to queue for the disk. Discrete event simulation using C-Sim was used to study the performance benefits of pre-fetching inlines.

The results in [15] show that significant reductions in response time can be achieved with reasonably small caches. Figure 2 shows the variation of response time of inlines versus cache size. The reduction in response time between a 0 and 128-KB cache is 46%, while the response time drops only 20% between 128 and 1024 KB.

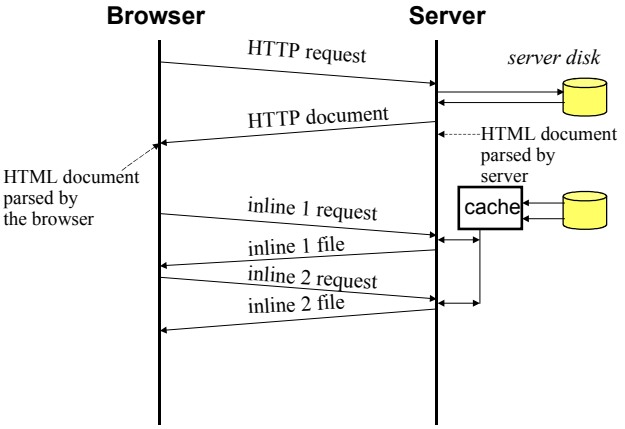

Figure 1 - Anatomy of a Web transaction with inline prefetching.

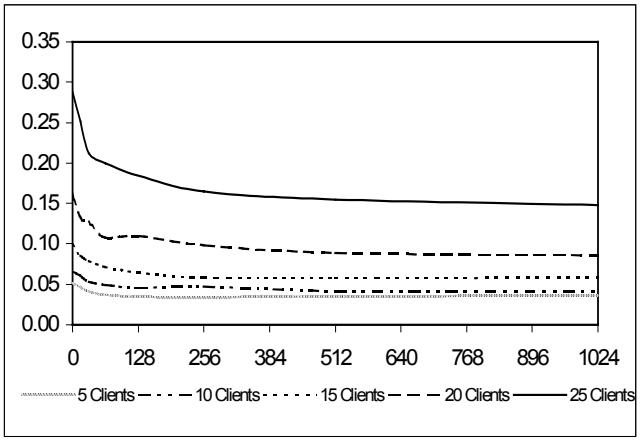

Figure 2 - Response time (in sec) of inline files vs. cache size (in KB)

4.2 Prefetching Results of Queries to Search Engines

Foxwell and Menascé [16] studied the potential benefits of prefetching results of queries to search engines. Users of WWW search sites, such as Yahoo and Lycos, receive dynamically generated Web pages that list URL "hits", or document links, whose contents match query keywords. Users then click on some or all of these links to inspect the document, at which time the server for the link transmits the file. In [16] the authors examined the behavior of users of Web search sites. Usage patterns were described for both simple and complex search queries; metrics were suggested for determining which search query results to prefetch in order to reduce network latency as seen by the client. Measurements suggested that for some search sites and user populations, the probability that a user will access a document is high enough to justify prefetching it. Figure 3 illustrates the estimated probability that a URL resulting

from a Lycos query will be followed. The x-axis indicates the order of appearance of the URLs on the result page.

The benefits of prefetching URLs resulting from queries to search engines can be characterized by the hit ratio of prefetched URLs as a function of the number *T*, the threshold, of results to be prefetched. This hit ratio is given by

$$HitRatio = \frac{\sum_{i=1}^{T} p_i}{1 + T + \sum_{i=T+1}^{N} p_i}$$

where p_i is the estimated probability that the i-th URL will be followed [16].

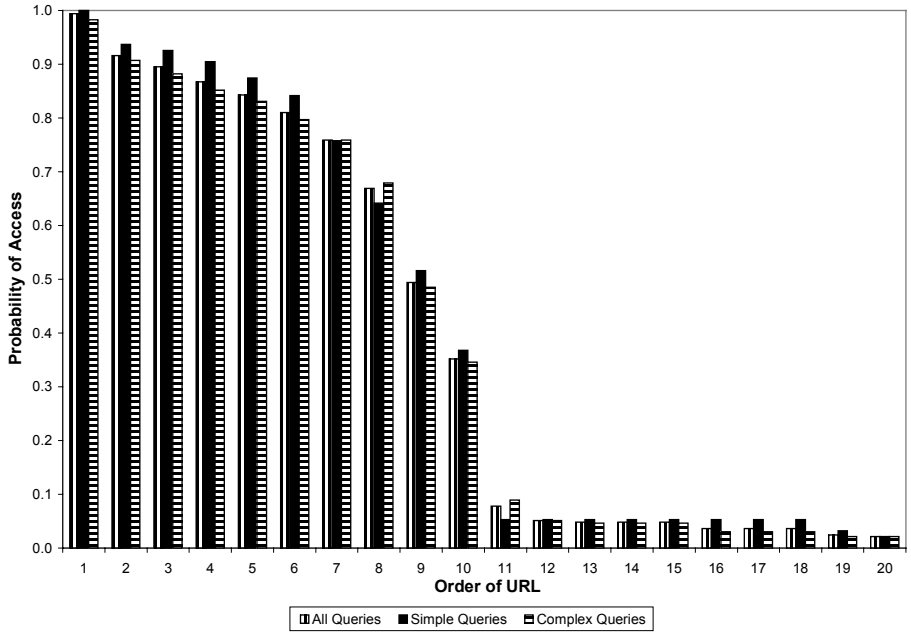

Figure 3 - Estimated Probabilities of Accessing URLs from Lycos Queries

Figure 4 shows the hit ratio versus the threshold T for Lycos queries. As it can be seen from the figure, the average hit ratio increases initially with the threshold *T*, achieves a maximum for an optimal threshold value *T**, and then decreases again. This indicates that a threshold value greater than *T** does not bring benefits in terms of higher hit ratios due to a small probability of the URLs being accessed.

Others have studied pre-fetching on the Web in different contexts. Padmanabhan studied the effects of prefetching files on latency, network load, and service to other users [22]. Crovella and Barford suggested that by adjusting the rate at which files are prefetched, the additional network load can be minimized [11]. Chinen and Yamaguchi developed a prefetching proxy server, and studied the effects of retrieving the first N links within *any* Web page viewed by a user [10].

5. Predicting Web Performance

Web performance can be predicted with the help of performance prediction models, which can be simulation or analytic-based. .*Simulation models* are computer programs that mimic the behavior of transactions and client/server requests as they move from one component of the system to the other. Simulation models tend to be quite detailed by their very nature, require detailed input parameters, and are generally very compute-intensive. Their main advantage is that they can model virtually all possible systems whereas analytic models may have some limitations due to the mathematical tractability of some situations.

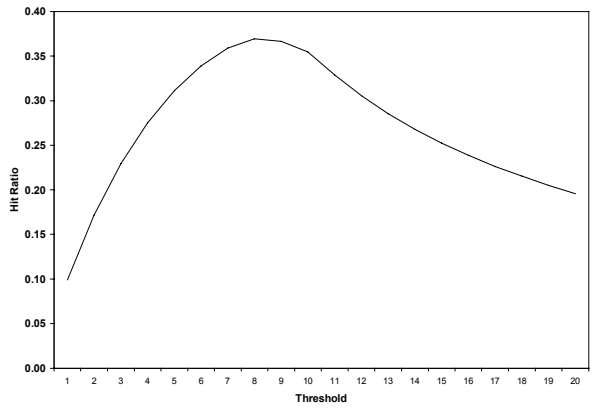

Figure 4 - Hit Ratio vs. Threshold for Lycos Queries.

Analytic models are based on a set of formulas and/or computational algorithms that relate system and workload parameters to performance metrics such as response times, throughputs, utilization, queue lengths, and connection rejection probabilities. Significant work has been done in the past twenty five years with respect to the development of efficient computational algorithms for solving queuing networks. These developments include the convolution algorithm developed by Jeffrey Buzen and the Mean Value Analysis (MVA) technique developed by Martin Reiser and Steve Lavenberg (see [18, 20] for details on MVA). Their work enabled the theory of performance modeling to become available to practitioners in the field and has spurred the development of many successful tools that use these techniques. Most contemporary modeling tools are based on the study of queuing theory and queuing networks (QNs).

The first results on solving QNs dealt with a special type of QN, called *product-form* QNs, for which an efficient exact solution could be found. Approximations had to be developed to deal with situations not contemplated by the so called, product-form solution [2]. These approximations enabled the successful modeling of situations in which parts of a system have a constraint on the maximum number of requests in the system. For example, the maximum number of threads of an HTTP server, or simultaneous resource possession cases, as required in the modeling of the persistent HTTP protocol.

QN models represent each resource by a *queue* composed of the resource and a waiting queue for the resource. A request may visit a given resource more than once during its execution. For example, an HTTP request may require several I/O operations on a given disk. The total time spent by a request obtaining service from a resource is called the *service demand.* Service demands do not include any queuing and constitute one of the important types of input parameters to QN models. In queuing network terminology, the entity that flows from one queue to another is a *customer*. Customers may represent HTTP requests, database transactions, remote file service requests. A customer is the model representation of the workload unit of work. Since a system may contain many different types of workloads contending for resources, a QN model may contain many different *classes of customers*. For example, different classes of customers in a QN may model requests for small text documents and large image files stored at a Web server, since they represent substantially different usage of server and network resources.

Customer classes can be further divided into two types: open and closed. *Open classes* are those where the total number of customers at any given time in the QN is not bounded. The *average arrival rate* of customers and the service demands at each queue characterizes an open class. Open classes model situations where the arrival rate of customers to the system is independent of the number of customers in the system. This is true of Web servers available to the public via the Internet; the customer population is very large and unknown. *Closed classes* are those where the total number of customers at any given time in the QN is restricted. The total number of customers in the class (*customer population*) and the service demands at each queue characterize closed classes. Closed classes model situations where the arrival rate of customers depends on the total number of customers in the system. An example is the Web server on the intranet of a company available only to the company's employees.

Existing QN solution techniques cannot handle exactly some of the important characteristics of Web workloads such as self-similar traffic and heavy tailed distributions of file sizes. However, one can adapt existing solution techniques [18, 20] to generate approximate models that reflect, to a certain level of accuracy, the impact on performance of these new Web workload features [19]. The following two subsections provide examples on how this can be achieved.

5.1 Effects of Burstiness on Performance

Section 3 showed that several studies concluded that Web traffic is bursty in nature. It was observed, through measurements of a Web server, that burstiness decreases the throughput of the server [7]. Menascé and Almeida captured this effect in analytic models of Web servers [19]. They defined two parameters, *a* and *b*, that characterize the burstiness of a workload:

- *a:* ratio between the maximum observed request rate and the average request rate during an observation period.

- *b:* fraction of time during which the instantaneous arrival rate exceeds the average arrival rate.

In [19], an operational approach is given to compute the values of parameters *a* and *b* from information contained in an HTTP log. Consider an HTTP log composed of L requests that arrive to a Web server during a time interval of τ sec. So, the arrival rate λ of requests during this interval is $\lambda = L / \tau$. Let the interval τ be divided into *n* subintervals of duration τ / n called *epochs*. As shown in [19], the parameters *a* and *b* can be computed as follows.

Let

- Arr (k): number of HTTP requests that arrive in epoch k,
- λ_k: arrival rate during epoch k computed as Arr (k) / (τ / n),
- Arr^+: total number of HTTP requests that arrive in epochs in which $\lambda_k > \lambda$, and
- $\lambda^+ = Arr^+ / (b\ \tau)$.

So,

b = (number of epochs for which $\lambda_k > \lambda$) / n and
$a = \lambda^+ / \lambda = Arr^+ / (b\ L)$.

The effect of burstiness on performance can then be captured by adding a burstiness term $\alpha\ x\ b$ to the fixed service demand D_f of the processor of a Web server. Hence, the effective service demand D is

$$D = D_f + \alpha\ x\ b$$

where the constant α is computed by measuring the utilization, the server throughput and the values of the parameter *b* for two consecutive subintervals of the total measurement interval. Thus,

$$\alpha = (U_1/X^1_0 - U_2/X^2_0) / (b_1 - b_2)$$

where U_1 and U_2 are CPU utilizations, X^1_0 and X^2_0 are throughputs, and b_1 and b_2 the values of the parameter *b* during the two consecutive intervals. A QN model for the Web server can now be solved using the effective service demand computed above.

Figure 5 depicts the variation of the throughput of a Web server versus the burstiness factor. The maximum throughput of 50 requests/sec is observed for non-bursty workloads (b = 0). Even for moderately bursty workloads (b= 0.3), the throughput is less than one third of the throughput for a non-bursty workload.

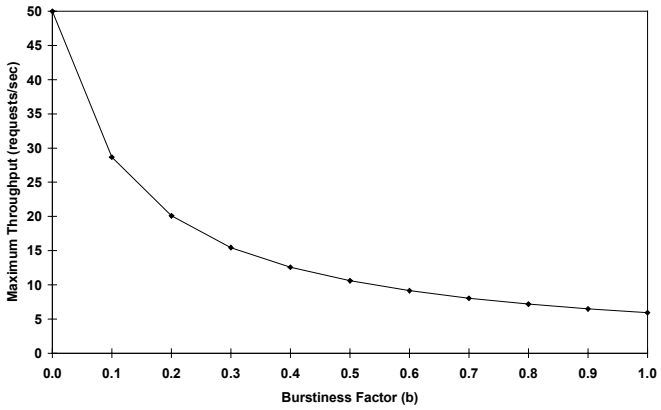

Figure 5 - Maximum Throughput vs. Burstiness Factor

5.2 Effects of Heavy-Tailed File Size Distributions on Performance

Another feature of Web traffic is that the size of documents retrieved ranges widely, from 10^2 to 10^8 bytes. While most of the documents tend to be small, there is a non negligible fraction of documents that are very large. So, file sizes and other important distributions of Web traffic exhibit heavy tails which implies in a very large variability in the values of the workload parameters. Table 2 gives an example of a typical distribution of file sizes retrieved from a Web site [19]. While 65% of the documents are smaller than 50KB, 5% are bigger than 500 KB. This is true because of the variety of objects stored at Web sites (e.g., html files, images, sound, and video clips).

Due to the large variability of the size of documents, average results for the aggregate collection of documents have very little statistical meaning. Categorizing the requests into a number of classes, defined by ranges of document sizes, improves the accuracy and significance of performance metrics. Multiclass queuing network models [19, 20] can be used to account for heavy tailed distributions of document sizes.

Table 2 - Typical Distribution of File Sizes Retrieved from a Web Site.

Class	File Size Range (KB)	Percent of Requests
1	size < 5	25
2	5 ≤ size < 50	40
3	50 ≤ size < 100	20
4	100 ≤ size < 500	10
5	size ≥ 500	5

6. An Example

Consider the intranet of figure 6 composed of various clients connected through a LAN. A router connects the LAN to the Internet through a serial link. On the same LAN there is a caching proxy server that holds the most recently fetched documents from external Web servers. Access to documents stored at the proxy cache (a cache hit) is much faster than access to documents stored at remote servers.

Suppose we want to determine the potential benefits of upgrading the connection to the Internet. The QN model of figure 7 can be used to represent the intranet and predict the performance under various different link speeds.

Table 3 shows the variation of the response time of HTTP requests and of the proxy server's throughput as the speed of the link to the Internet varies from 56 Kbps to 1.5 Mbps (a T1 link). One can see the dramatic improvement in response time (from 9.996 sec to 0.984 sec) as the link speed varies from 56Kbps to 256 Kbps. In other words, a ten-fold decrease in response time was achieved with an approximate four times increase in the speed of the link to the Internet. It can be seen also that the advantages of upgrading the link to T1 speeds are not that big because after some point, the link is no longer the bottleneck.

The example shows the value of performance models to predict the performance of Web servers and intranets and help in making informed decisions regarding the adequate sizing of Web infrastructures.

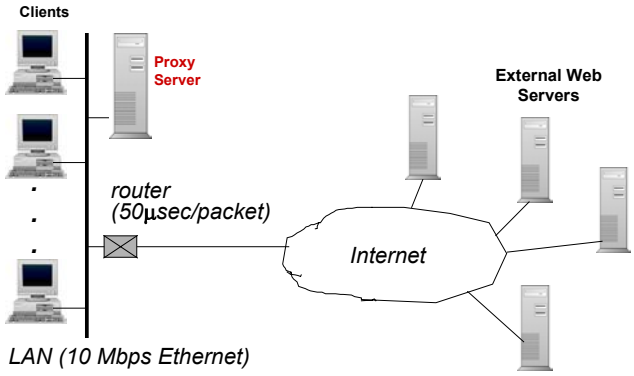

Figure 6 - An Intranet with a Proxy Server.

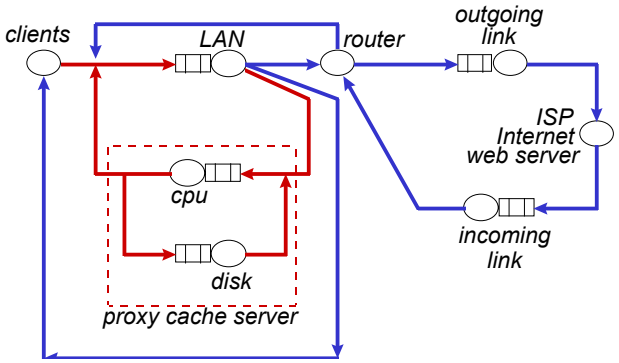

Figure 7 - QN model corresponding to the intranet of figure 6.

6. Concluding Remarks

The Web is becoming an important infrastructure for electronic commerce. The potential savings of e-commerce will only be achieved if the Web delivers the performance needed to support mission-critical applications. This paper discussed the growth of e-commerce, the characteristics that make WWW traffic different from other types of workloads, examples of how caching and prefetching can be used to improve performance, and the type of performance models that can be used in sizing Web servers and intranets.

Table 3- Performance Results for the Intranet Example

Link Bandwidth (Kbps)	Throughput (HTTP req/sec)	Response Time (sec)	Bottleneck
56	1.125	9.996	Link to the ISP
128	2.393	2.935	ISP + Internet + External

			Web Server
256	3.475	0.984	ISP + Internet + External Web Server
1500	3.883	0.530	ISP + Internet + External Web Server

References

[1] Abrams, M., C. Standridge, G. Abdulla, S. Williams, and E. Fox, "Caching Proxies: Limitations and Potentials," World Wide Web Journal, Vol. 1 (1).

[2] Agrawal, S., Metamodeling: *A Study of Approximations in Queuing Models*, MIT Press, 1984.

[3] Almeida, V., A. Bestavros, M. Crovella, and A. Oliveira., "Characterizing Reference Locality in the WWW," Proc. PDIS'96: The IEEE Conference on Parallel and Distributed Information Systems, Miami Beach, FL, IEEE.

[4] Almeida, V. A. F. and A. Oliveira, "On the fractal nature of WWW and its applications to cache modeling," Technical Report TR-96-004, Boston University, CS Dept, Boston, February 5, 1996.

[5] Arlitt, M. and G. Williamson, "Web Workload Characterization," Proc. of the 1996 SIGMETRICS Conf. Measurement Comput. Syst., ACM, Philadelphia, May 1996.

[6] Abrams, M., Standridge, C., Abdulla, G., Williams, S., and Fox, E. "Caching Proxies: Limitations and Potentials," *The World Wide Web Journal* 1(1), 1995.

[7] Banga, G. and P. Druschel, "Measuring the Capacity of a Web server," USENIX Symposium on Internet Technology and Systems, Dec. 1997.

[8] Bray, T., "Measuring the Web," *The World Wide Web Journal*, 1 (3), 1996.

[9] Business Week, June 22, 1998.

[10] Chinen, K., and Yamaguchi, S. "An Interactive Prefetching Proxy Server for Improvements of WWW Latency," Nara Institute of Science and Technology, 1996.

[11] Crovella, M. and P. Barford, "The Network Effects of Prefetching," Computer Science Department, Boston University, 1997.

[12] Crovella, M. and A. Bestravos, "Self-Similarity in World Wide Web Traffic: evidence and possible causes," Proc. of the 1996 SIGMETRICS Conf. Measurement Comput. Syst., ACM, Philadelphia, May 1996.

[13] Coffman, Jr., E. and P. J. Denning, *Operating Systems Theory*, Prentice Hall, Upper Saddle River, 1973.

[14] Cunha, C., A. Bestavros, and M. Crovella, "Characteristics of WWW Client-based Traces," Technical Report, TR-95-010, Boston University, CS Dept, Boston, MA 02215, April, 1995

[15] Dodge, R. and D. A. Menascé, "Prefetching Inlines to Improve Web Server Latency," Proc. 1998 Computer Measurement Group Conference, Anaheim, CA, Dec. 8-11, 1998.

[16] Foxwell, H. and D. A. Menascé, "Prefetching Results of Web Searches," Proc. 1998 Computer Measurement Group Conference, Anaheim, CA, Dec. 8-11, 1998.

[17] Gillmor, D., "The Art of Internet Commerce: Dell Computer Corporation," *Hemispheres*, June 1998.

[18] Lazowska, E. D., J. Zahorjan, G. S. Graham, and K. C. Sevcik, *Quantitative System Performance: Conmputer System Analysis Using Queuing Network Models,* Prentice Hall, Upper Saddle River, NJ, 1984.

[19] Menascé, D. A. and V. A. F. Almeida, *Capacity Planning for Web Performance: metrics, models, and methods,* Prentice Hall, Upper Saddle River, 1998.

[20] Menascé, D. A., V. A. F. Almeida and L. W. Dowdy, *Capacity Planning and Performance Modeling: from mainframes to client-server systems,* Prentice Hall, Upper Saddle River, 1994.

[21] M. Mendes and V. A. F. Almeida, "Analyzing the Impact of Dynamic Pages on the Performance of Web Servers," Proc. 1998 Computer Measurement Group Conference, Anaheim, CA, Dec. 8-11, 1998.

[22] Padmanabhan, V., "Improving World Wide Web Latency," Computer Science Division, University of California at Berkeley, 1995.

[23] Pitkow, J., PARC HTTP-NG: Web Characterization Reading List, http://www.parc.xerox.com/istl/projects/http-ng/web-characterization-reading.html

[24] Tauscher, L. and S. Greenberg, Revisitation Patterns in World Wide Web Navigation, Proc. ACM SIGCHI'97 Conference on Human Factors in Computing Systems, Atlanta, GA, 1997.

[25] USA Today, June 18, 1998.