Web Facts and Fantasy

Stephen Manley, *Network Appliance*
Margo Seltzer, *Harvard University*

Abstract

There is a great deal of research about improving Web server performance and building better, faster servers, but little research in characterizing servers and the load imposed upon them. While some tremendously popular and busy sites, such as netscape.com, playboy.com, and altavista.com, receive several million hits per day, most servers are never subjected to loads of this magnitude. This paper presents the analysis of internet Web server logs for a variety of different types of sites. We present a taxonomy of the different types of Web sites and characterize their access patterns and, more importantly, their growth. We then use our server logs to address some common perceptions about the Web. We show that, on a variety of sites, contrary to popular belief, the use of CGI does not appear to be increasing and that long latencies are not necessarily due to server loading. We then show that, as expected, persistent connections are generally useful, but that dynamic time-out intervals may be unnecessarily complex and that allowing multiple persistent connections per client may actually hinder resource utilization compared to allowing only a single persistent connection.

1. Introduction

The public's enthusiasm for the Web has been matched only by that of computer companies. Since money drives the industry, the emphasis from Microsoft to the smallest start-up has been on rapid technological development, rather than well-reasoned scientific advancement. New Web tools make money, statistical analyses of the Web do not. Concurrently, the academic community is often constrained by lack of data from sites outside the academic world [1][2][6]. As a result, most existing analyses are outdated. Most published statistics on server behavior come from data gathered in late 1994. In the past three years, the nature of the Web has fundamentally changed. CGI has introduced server and user interaction. Java and animated GIF images have continued to find their way onto Web pages. The number of Web users has grown at an unknown, but predictably exponential rate. With the tremendous growth and change in Web sites, users, and technology, a comprehensive analysis of real traffic can help focus research on the most instrumental issues facing the computing community.

The statistical analysis presented in this paper focuses on traffic patterns observed on a variety of Internet Web sites (intranet servers have been omitted from this study due to the unavailability of intranet server logs). Server logs reveal an enormous amount of information about users, server behavior, changes in sites, and potential benefits of new technical developments. In order to design next generation services and protocols (e.g., HTTP-NG), it is crucial to understand what the Web looks like today, how it is growing, and why it is growing as it is. In this paper, we use internet Web server log analysis to dispel and confirm widely held conceptions about the Web. Section 2 describes the initial set of sites we surveyed to derive our Web site taxonomy and outlines the basic growth characteristics of these sites. Section 3 presents a simple taxonomy for describing site growth, dispelling the myth that all Web sites are alike. Section 4 dispels the myth that CGI traffic is becoming uniformly more important. Section 5 addresses the issues surrounding persistent connections and how to maximize the benefit derived from them. Section 6 takes a first step towards answering the question, “What makes users wait?” by showing that servers are not necessarily the primary source of latency on the Web. Section 7 concludes.

2. Site Survey

Our Web log analysis is based on server logs obtained from a variety of sites. The sites were chosen to cover a broad range of topics, user populations, popularity, and size. Due to our agreements with several of our providers, we are unable to identify the sites in question, so we provide descriptions of the sites instead. Table 1 summarizes our initial set of server logs.

The sites in our survey can be broadly described in three basic categories: academic sites (EECS, FAS, ECE), business (BUS, ISP, AE, WEB, FSS), and informational (PROF, GOV). Within these categories, the sites exhibit different characteristics. For example, the business sites represent each of the major business models on the Web. ISP is an internet service provider that advertises its services. BUS also uses the Web for advertising, but its business has nothing to do with the Web. AE comes from the set of ubiquitous adult-

Abbr	Anonymized Site Name	Site Type	Content	Service Provider	Server Software	Time
BUS	Traditional Business	.com	Information on subject matter, advertisements	ISP	Apache 1.1.3	1/96 - 2/97
EECS	Harvard University Electrical Engineering and Computer Science	.edu	Graduate student Web pages, department information	Harvard EECS	NCSA 1.4.2	4/96-2/97
FAS	Harvard University Faculty Arts and Sciences	.edu	Information for an academic institution, student Web pages	Harvard FAS	NCSA 1.4.2 Apache 1.1.3	10/94-2/96
ISP	ISP company page	.com	Simple advertisement	ISP	Apache 1.1.3	9/96 - 2/97
ECE	Rice University Electrical and Computer Engineering	.edu	Graduate student Web pages, department information	Rice ECE	Netscape Netsite-Commerce 1.0	7/95-12/96
AE	Adult-Entertainment	.com	Adult images, movies, chatrooms	ISP	Apache 1.1.3	3/96-9/96
PROF	Organization for Members of same Profession	.org	Articles and images pertaining to field	ISP	Microsoft IIS/3.0	4/96-2/97
WEB	Web site designer	.com	Samples of different sites, games	ISP	Apache 1.1.3	8/96-2/97
GOV	Government Agency	.gov	Information on agency's actions	ISP	Apache 1.1.3	8/96-2/97
FSS	Free Web Software Site	.com	Evaluation copy of proprietary Web software	ISP	Apache 1.1.3	4/96-1/97

Table 1: Site Survey Description. The educational sites (EECS, FAS, and ECE) differ from the rest of the sites in that the sites' content is not the uniform product of a single webmaster, but instead a conglomerate of a number of independent Web publishers.

entertainment sites. FSS makes its living by licensing a Web software product, but allowing visitors to download a less functional version of the product for free. The final business model is that of WEB, which uses its site as an advertisement for a Web product. Although all the ISP logs in Table 1 come from a single ISP, we have analyzed logs from other providers and found that these sites are indicative of the other providers' sites as well.

Unsurprisingly, the characteristic common to nearly all our sites is an exponential rate of change. Table 2 shows this change in requests, bytes transferred, number of files on the sites and the number of bytes on the site for each of our surveyed sites. While the derivative of the change for three of our sites is negative, perhaps the most astounding result is that even the slowest growing sites double each year and our fastest growing site doubles each month.

3. A Web Site Taxonomy

Much Web research tends to assume that all interesting sites have traffic loads similar to those of Microsoft and Netscape. These sites each have more than ten servers to handle tens of millions of requests each day, claiming to be two of the most popular on the Web. However, as the most heavily loaded sites, they cannot also be the common case. While most of the sites in Table 2 demonstrate substantial growth, the loads, shown in Table 3, vary tremendously. A site handling fifteen million requests for seventy-six GB of data per month (FSS) must be thought of differently than a site processing forty-five thousand requests for 250 MB of data per month. Comparing these sites directly is unlikely to yield very interesting results. Load is only one way in which sites differ; the size of the site, the diversity of the population that is attracted to the site, the growth patterns, the user access patterns, and how the site changes all play large roles in characterizing a Web site. From our log analysis, we have concluded that the three primary issues that characterize a site are: site composition and growth, growth in traffic, and user access patterns. While the data for the first two factors can be found in Table 2 and Table 3, user access patterns are not easily described. The distribution of requests per file and distribution of number of requests per user indicates whether users tend to visit many pages on a site, or only a few. These figures also indicate whether all users visit the same subset of pages, or tend to view different subsets of pages on the site. We present a more detailed analysis of these phenomena in earlier work [7].

Site	% Growth per month for duration of logs.				Double (Half) Interval
	Reqs	Bytes	Files	Bytes	
			on Site		
Traditional Business	60	105	37	67	2 months
Harvard EECS	28	18	19	16	3 months
Harvard FAS	27	31	33	33	3 months
ISP	-2	7	-2	9	3+ years
Rice ECE	13	17	7	14	6 months
Adult Entertainment	-27	-29	1	-1	3 months
Organization	-21	-20	-23	-19	3 months
Web Site Designer	6	7	0	14	1 year
Government Agency	7	5	1	-7	11 months
Free Software	95	81	23	24	1 month

Table 2: The monthly growth patterns of each site and its traffic. As the growth for nearly all these sites is exponential, the interesting question becomes, “How long does it take to double?” As we can see by the Free Software Site, there are examples of sites that nearly double every month while other sites (Web Site Designer) grow more slowly. Some of the sites actually demonstrate negative growth, another frequent Web phenomenon that will be discussed in Section 3. In particular, the Adult Entertainment site no longer exists. The reported data traces its reduction to destruction.

Site Name	Requests	Monthly Transfer Rate (MB)	Files on Site	MB on Site
Traditional Business	321,747	3,819	347	2.8
Harvard EECS	106,001	1,322	5,865	196.0
Harvard FAS	2,328,401	15,097	34, 348	455.0
ISP	8,139	39	134	1.5
Rice ECE	85,763	854	4,655	115.0
Adult Content	69,906	857	223	5.5
Organization	42,301	251	95	0.8
Web Site Designer	43,523	104	119	0.7
Government Agency	26,049	214	185	1.2
Free Software	15,982,085	76,315	4070	136.0

Table 3: The size of sites and the traffic handled in the most recent server log (one month). The disparity in levels of traffic and site size illustrate the fundamental difference in Web sites.

Growth Function	Site(s)	Explanation
# of Web Users	Single topic sites: FSS	More users learn of the site and visit it to download software.
Site overhaul	Aggressive business advertising: BUS	Grow in bursts as the webmasters “renovate” the site.
Number of documents on site	Academic sites: EECS, FAS, ECE	Have a disproportionate number of pages, and their popularity increases as more users create pages on the site.
Documents visited per user	Non-aggressive businesses or special interest sites: ISP, GOV, WEB	Lure visitors into visiting more of the site as it develops.
Number of search engine hits	Competitive markets: AE	Grow based on number of times the popular search engines find them.
Cost	Pay-for-View Sites: PROF	Increasing fees are met with decreasing traffic

Table 4: Characteristics that Categorize a Web Site. The “growth function” column identifies the parameter that most closely correlates to a site’s growth. We hypothesize that FSS is representative of a very large class of sites whose popularity grows with the user population of the Web. Search engine sites and sites for general entertainment and information (CNN, ESPN, etc.) are hypothesized to fall in this category as well.

During the course of our monitoring of these sites, we visited each site frequently to determine how the sites were evolving and then used that information in conjunction with the logs to discern basic trends in site growth. We conducted a regression analysis on the growth of the site (as measured by the number of requests) for every parameter we could measure. We found that many parameters appeared to have a slight influence on growth, but we focused on that parameter that correlated most closely with growth. For some sites, the parameter attributed to growth showed excellent correlation (e.g., better than 95% confidence intervals for sites such as AE). For other sites, the best parameter produced 80% confidence intervals (e.g., FSS). In all cases, the best parameter provided confidence intervals of at least 80%. Table 4 summarizes the growth patterns.

There are a variety of ways in which sites can grow. They may grow because new users are drawn to the site or because existing users visit more frequently or more deeply. Our first class grows by attracting more visitors to the site, and we speculate that the number of visitors is a function of the total Web user population. The free software site has a singular, wildly popular product. As more people learn of the software, more people visit the site to download the software. The accesses on this site are heavily skewed: 2% of the documents account for 95% of the site’s traffic.

A second growth model is to explicitly renovate a site in an attempt to increase traffic. The business using the Web to market aggressively (BUS) demonstrates this

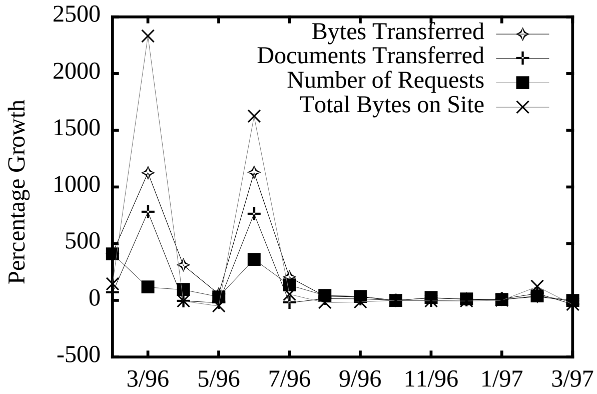

Figure 1: Traditional Business Site Growth. The two dramatic peaks indicate periods where the site was overhauled. Each time, the site's traffic exhibited a significant increase and then leveled off at this new level.

growth. Figure 1 depicts the site's growth during our evaluation interval. All growth occurs in bursts, whose timing corresponds to major reorganizations of the site. The first burst corresponds to a shift from a text-based site to a graphics-based site. The second burst corresponds to adding more depth to the site, adding more details about specific products and catering to particular classes of customers (e.g., women or young adults). After each reorganization, the site undergoes tremendous growth, which tapers off, and levels out at a volume that is significantly greater than it was before the renovation.

The third category is typified by the academic sites. The content of these sites is not controlled by a single Web master. Instead, it tends to grow with the user population; as the site grows in size, so do the number of requests to that site.

The fourth classification contains those sites whose traffic increases by attracting users to visit more of the site. There are two discernible patterns that characterize these sites. First, the number of requests closely tracks the number of documents on the site. Second, the average number of pages visited per session also tracks the growth of the site, as shown for the GOV site in Figure 2. The sites rarely change scope, but additional material is added on specific subjects, and the visitors respond by viewing a larger fraction of the site.

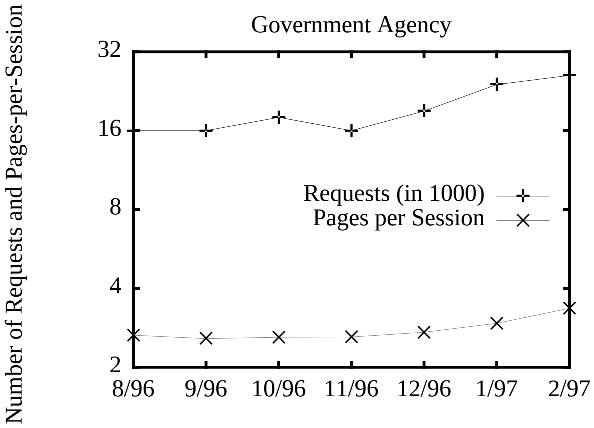

Figure 2: Growth tracks the number of pages visited per session. This shows the Government Agency where the gentle growth in number of requests stems from a similar growth in the number of pages visited by each user.

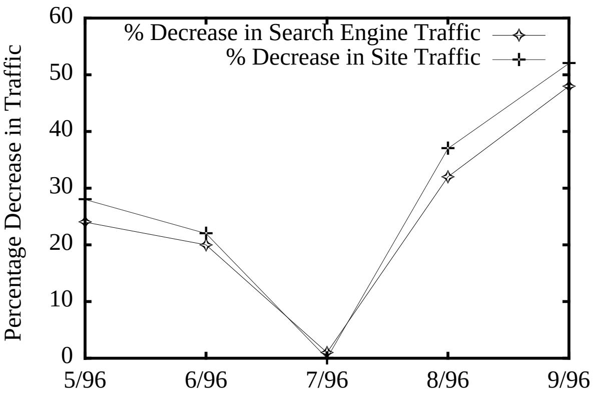

Figure 3: Growth of Search Engine Hits and User Requests for the Adult Entertainment Site. Those sites in a competitive market, such as the adult entertainment market, can live or die by their ranking in the various Web search engines.

The final two classes exhibit negative growth. First, consider the case of the Adult Entertainment site, which no longer exists. The business model of the site is like many on the Web—the user is given access to a limited subset of free material followed by a request for payment to get access to the remaining material. With the tremendous growth of the Web, and almost ubiquitous nature of such sites the best means of attracting users is through the search engines. Not surprisingly, the site experienced growth of almost a factor of 60, when it began to receive requests that were traced to the most commonly used search engines (e.g., Alta Vista, Yahoo, and Excite). The site's popularity began to decrease, however, without the site or user access patterns changing drastically. The number of unique users began to drop. Figure 3 shows that the number of user requests mirrors that of the number of references from the Web's search engines. Those sites that depend on search engines for rapid growth can also suffer a rapid downfall, when the search engine does not return the site's URL as one of the best matches.

Our final category also depicts negative growth. In this case, the site's popularity dropped off by an order of magnitude as soon as it began charging for access, and then began to exhibit growth similar to that of the non-aggressive business sites; those users who stuck with the site even after the site began to charge for access increased their use of the site, viewing an increasing fraction of it over time.

After completing the taxonomy based on our first set of logs (those shown in Table 1), we analyzed a set of logs from a second ISP to verify if they too fit into our taxonomy. The second ISP's sites all fell into two of our six categories: 15 of 28 sites grew by attracting more users and 13 of the 28 grew by encouraging users to visit the site more thoroughly, thus viewing more pages per session.

There is a class of sites that have been omitted from this survey, namely the Web search engines. We hypothesize that these sites fall into our existing taxonomy with respect to growth, in that their load reflects the growth of the Web in general (as does the Free Software Site or sites whose load grows as a function of the number of different visitors). However, they do not fall within the following discussion about the importance of CGI. It is obviously the case that sites hosting search engines will be extremely sensitive to CGI performance (or whatever they use to implement searching capabilities).

4. The Effects of the Common Gateway Interface (CGI)

Because processing CGI is frequently much more computationally expensive than returning static documents, its perceived importance has motivated a great deal of server development. Both Netscape and Microsoft have changed the interface for CGI traffic, to improve performance. Microsoft's site includes two servers dedicated to processing their equivalent of CGI. Yet, with all of the clamor, the sites we surveyed derived little functionality from CGI. Table 5 shows that of the servers we surveyed, most process very little CGI traffic. In fact, only three sites report more than 2% of their traffic due to CGI.

Of the sites we surveyed, the most widely used CGI script was the ubiquitous counter (a simple CGI script that tallies the number of accesses to a particular page) and the second most frequently occurring script was the redirect, a script that indicates that a page has moved. Although CGI, in general, is often assumed to be an order of magnitude slower than returning static HTML documents, these particular instances of the use CGI require about as much processing overhead as static documents [7]. The Adult Content site, Free Software Site, and Organization site also use CGI scripts to allow users to log into the site, and the Traditional Business and FAS sites provide search engine capabilities, which are responsible for a noticeable fraction of their CGI traffic. Even so, these hits account for a tiny fraction of the traffic on all but the organizational site, which is rather unusual in that all external requests are directed through a CGI-driven interface. The other anomalous site is EECS where students have access to the CGI bin and can create their own scripts. This site exhibits the greatest diversity in CGI and explains the relatively large percentage of traffic (and bytes) due to CGI (see `http://www.eecs.harvard.edu/collider.html` for a particularly creative use of CGI scripts). Perhaps most interestingly, we find that, not only is the use of CGI fairly low across all sites, but the percentage of traffic due to CGI did not increase over the course of our measurement interval.

Site Name	%Requests as CGI	%Bytes Transferred from CGI
Traditional Business	1.0	0.4
Harvard EECS	8.0	15.0
Harvard FAS	1.4	1.6
ISP	0.0	0.0
Rice ECE	0.0	0.0
Adult Content	2.0	0.0
Organization	34.0	62.0
Web Site Designer	1.0	0.0
Government Agency	0.0	0.0
Free Software	10	5.0

Table 5: The percent of requests due to CGI. Most servers process very little CGI, and the traffic it generates accounts for a small fraction of the site's traffic.

In the logs we have examined, the latency of CGI requests has mirrored that of regular requests, and we find that sites with significantly different ratios of CGI to non-CGI requests exhibit the same latency patterns. Based on this observation and the fact that the ratio of CGI traffic to regular traffic is not changing, we conclude that the long latencies users are experiencing at these sites or any increased slowdowns of these sites is not due to CGI. Section 6 presents a more detailed discussion of observed latencies.

5. Persistent Connections

The HTTP/1.1 specification [4] calls for support of persistent connections; that is, rather than initiating a new connection for every document retrieved from a server, a long-lived connection can be used for transmitting multiple documents. Initial research in this area demonstrated that for two sites under analysis (the 1994 California election server and a corporate site), if connections were held open for 60 seconds, then 50% of the visitors to the site would receive at least 10 files per open connection. On average, each connection supported six requests, most connections were reused, and yet the number of open connections remained low [9]. Later analysis by the World Wide Web Consortium [10] showed that the current practice of maintaining parallel open connections (e.g., four connections for the Netscape browser) was crucial for achieving acceptable latency. If we apply these results to the persistent connection issue, then it's possible that it is necessary to maintain multiple persistent connections per session. We wanted to investigate the resource utilization effects due to maintaining multiple persistent connections per session.

Using a simulation based on a subset of our server logs, we explored four persistent connection parameters: the time-out interval, the maximum number of connections allowed per user, the maximum number of open persistent connections, and the algorithm for implementing dynamic time-out. Our simulator has three characteristics that detract from the behavior a server would observe in reality. First, we assume that each IP address corresponds to a single user, and therefore, can create only one session with a server. Second, the logs are biased toward browsers that make four concurrent connections, the standard Netscape browser behavior. Such a bias makes it impossible to accurately predict the user-perceived latency that will result when considering only one or two simultaneous connections. Third, in one of our logs (Harvard FAS), the server does not record the latency between receiving a request and sending a response. In these cases, the simulation assumes 0-request-handling latency. Therefore, the FAS results will tend to be pessimistic about the effectiveness of persistent connections. The pessimism occurs when we assume data has been transmitted instantaneously and use that time as the "last active" time of the connection. In reality, the latest activity on that connection will occur after the response has been handled, and the possibility exists that we will time the connection out prematurely. Similarly, the estimations on the number of concurrently open connections for this site will tend to be small. In contrast, when we do have these latencies, then the simulation's time-out mechanism behaves exactly as a server's time-out mechanism. That is, the server begins the time-out period calculation as soon as it sends data over the connection, even though the client may receive the data much later, so the server's perception of how long a connection is idle may be significantly different from the perception of the client. While potentially suboptimal, this is the only knowledge that the server has, so it is used in timing out connections.

Name of site	Date simulated	Time period	Number requests
Traditional Business	2/28/97	24 hours	11,549
Harvard FAS	2/28/96	4 hours	16,741
Free software	2/28/97	1 hour	26,574

Table 6: Persistent Connection Simulation Data.

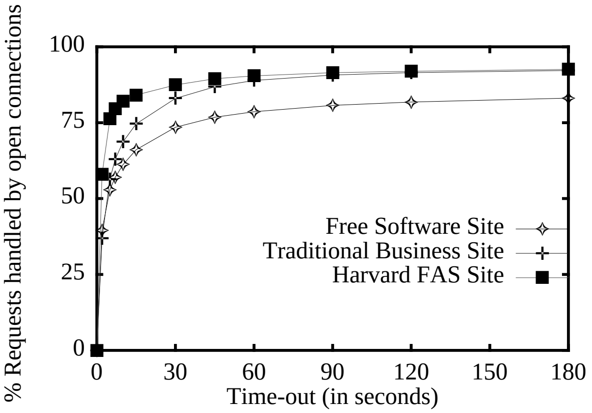

Figure 4: Sensitivity to Time-Out Intervals. In this simulation, we impose no maximum time-out limit, recording what percentage of requests could be handled by already open connections, as a function of the time-out interval. There is little additional benefit derived from leaving connections open longer than 15 seconds.

Table 6 describes the logs used, the dates and time periods that were run, and the number of requests processed. Although we chose our most heavily accessed sites, the time periods for each site differ because the levels of traffic vary so greatly. For each site, we selected four sets of each time period; we present the results of a single time period, but the results presented here are indicative for all the time periods.

We first ran the simulator with an infinite time-out interval, so we could determine the maximum benefit of persistent connections. In this simulation, the clients used only one persistent connection, generating the highest degree of connection reuse.

Figure 4 shows the persistent connection utilization as a function of the time-out interval. While the percentage of requests handled by persistent connections climbs rapidly up to a 15-second time-out, it remains relatively stable for intervals longer than 15 seconds. The greatest benefit for persistent connections

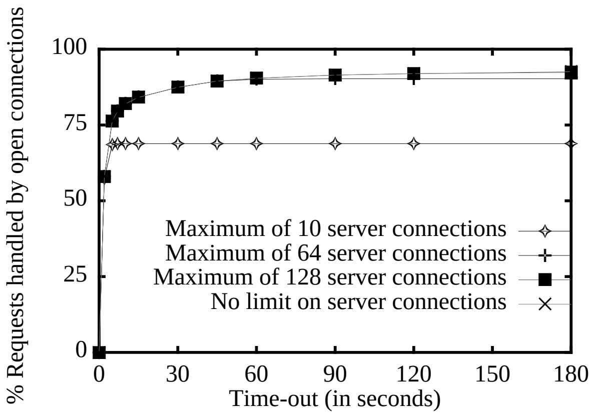

Figure 5: Sensitivity to Limiting the Number of Open Connections. This data shows the Free Software Site and the percentage of requests handled by persistent connections as we limit the maximum number of open connections. The connection limit can force connections to close prematurely reducing the benefit of the persistent connections.

occurs when users download a few pages, and these pages all use the same connection. On the FSS site, the majority of users exhibit this access pattern, so longer time-outs produce little benefit. Although users on the BUS site download more documents, they also derive little benefit from increased time-out intervals. Leaving connections open for longer intervals leads to a linear increase in the number of open connections. In fact, during the simulation of the site with the heaviest traffic (FSS), a three minute time-out resulted in over 300 open connections.

A more realistic analysis of persistent connections requires setting a limit on the number of open connections. Mogul focused his analysis on 10, 64, 128, and 1024 open connections. Our data shows that none of the servers opened as many as 1024 connections. In fact, only FAS and FSS ever had as many as 64 concurrently open connections. For the remainder of this discussion, we focus on FSS, because its logs include latencies and its heavier traffic enables a better analysis of the stresses that could significantly affect persistent connections.

Not surprisingly, the data show that, regardless of the connection limit and level of traffic, closing the least recently used connection leads to the best performance. Similarly unsurprising, whenever the time-out length leads to more active connections than permissible, increasing the time-out interval provides no improvement, because leaving the connections open longer exacerbates the situation, causing connections to be closed due to the imposed resource constraint, see Figure 5. Dynamic time-outs introduce no discernible benefit, because they effectively implement shorter static time-outs. The key insight is that the time-out interval and maximum open connections must be well-balanced. If fewer open connections are allowed than are necessary for the time-out interval, then connections will be closed prematurely. If more connections are allowed than the time-out interval warrants, the connections will be underutilized, wasting resource.

The second question we examined was how many persistent connections should be allowed per client. The HTTP/1.1 standard allows for up to two persistent connections per client [5], but we observed better resource utilization when clients are limited to a single persistent connection. The results presented in Figure 4 and Figure 5 were for a single persistent connection per client. Figure 6 shows what happens as we allow clients to have multiple persistent connections. The interaction of the number of persistent connections per client and the maximum number of open connections on the server results in worse resource utilization than might have been expected. Since the site is heavily loaded, allowing two connections per user doubles the number of open connections on the server. Therefore, at limits of 10 and 64 connections, the server closes connections more quickly than in the original model. As discussed before, this behavior has adverse effects on resource utilization. When we compare the resource utilization of the server allowing one persistent connection per user and 64 total connections to that of the server with two persistent connections per user and 128 total connections in Figure 6, we see the mild difference that we expect. Of course, such decisions do not come without cost. ISPs charge customers for extra connections; the implications of requiring the server to retain twice as many open connections have serious ramifications for the cost structure of service provision. Allowing two connections per client requires that servers potentially double the number of simultaneous open connections to achieve the high connection re-use rates we see in the one connection case. Unfortunately, at this point, we do not have enough data to incorporate the effect that more connections have on user response time.

6. Why Clients Wait

Long delays on the Web are often attributed to “overloaded servers,” and researchers have cited four causes of server latency: the number of TIME_WAIT network connections, the number of concurrently active requests, the cost of CGI, and the sizes of the files requested [3][8]. Server logs provide rather incomplete latency measurements, but we can use the information available to determine that users accessing the servers we analyzed do experience long latencies that cannot be

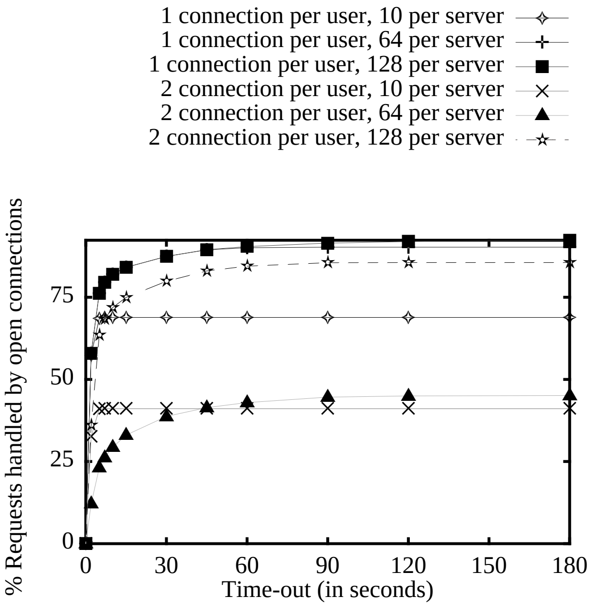

Figure 6: The Impact of Multiple Persistent Connections per Client. Allowing a client to create two persistent connections leads to the premature closing of many connections and a degradation in performance.

attributed to the server. The latency logged by the server includes the time between the server initially receiving a request and the server issuing the last 8 KB write in response to the request. In particular, this time does not include connection setup (which happens before the server gets the request), the time to transmit the last block of data, or the effect of virtual hosting (supporting multiple Web sites on a single machine). Nonetheless, given the albeit limited data in server logs, we are still able to determine that, even for our most heavily accessed site (FSS), the server was not responsible for any user-perceptible latency.

For this analysis, we chose 15 minute segments of near-peak activity on three of the servers, representing three different orders of magnitude of traffic. For the purpose of this discussion, we will focus on the most heavily used server (FSS). During the peak interval, the server handled 6722 requests which equates to a server handling approximately 650,000 requests per day. This site is the most heavily used site hosted by our first ISP, which is one of the largest ISPs in the country.

The server for this site breaks requests into 8 KB chunks, waiting until each chunk has been acknowledged before sending the next one. On the last chunk, the server considers its job done as soon as it writes the data into its network buffers. Our first step was to analyze all requests smaller than 8 KB, in which case, the latency recorded by the server is exactly the time the server spent handling the request. Even during periods of heavy activity, all such requests were handled in less than one second, and 50% of the requests were handled in less than 1 ms. So, for small files, the server is not introducing the latencies that plague users of the Web.

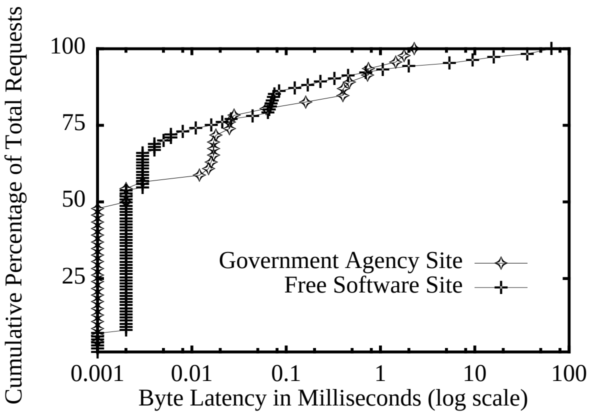

Figure 7: Byte Latencies for the Free Software Site (6722 hits/15 minutes) and the Government Agency Site (368 hits/15 minutes). Despite handling nearly 20 times as many requests as the Government Agency Site, the Software site shows similar byte-latencies.

For files larger than 8 KB, the latencies reported include the server overhead in addition to network and client delays. For our busiest site (FSS), these latencies range from 50 ms to 10 minutes; this is the time that users actually wait. We cannot directly compare these latencies because documents vary significantly in size, and we expect that it will take markedly longer to transfer a one megabyte file than a one kilobyte file. In order to analyze these requests accurately, we use the metric *byte-latency*, which is the average time that it takes the client to receive a single byte of data. Interestingly enough, when we compare byte-latencies across sites with loads that differ by more than an order of magnitude (FSS and GOV), Figure 7 shows that the distribution of byte-latencies is nearly identical and varies over four orders of magnitude; this kind of variation cannot be explained by any of the commonly proposed theories of server latency.

One common perception is that load and the number of open connections cause excessive server latencies. We assume that the number of active requests corresponds to server load, and examined the relationship between latency and both the average and maximum number of concurrent requests serviced while a request was being handled. Neither the average nor maximum shows any correlation to the byte-latencies.

Next we turn to the perception that CGI traffic is a cause of significantly increased latencies. However, during our intervals of peak activity, none of the CGI requests generated a response larger than 8 KB, and as mentioned earlier, none of the requests smaller than 8 KB required excessive processing time on the server.

Finally, we looked for a correlation between the size of the transfer and the byte-latency induced. Once again, there was no correlation.

From this series of analyses, we conclude that while some clients did observe long latencies from these servers, the latencies cannot be explained by server over-loading. The server has no difficulty handling most requests in under one ms, and the data from the server logs shows no indication that load, CGI, or file size contribute to the unpleasant latencies that users experience. We do find that the byte-latencies remain relatively fixed for given clients over 5, 10, and 15 minute intervals leading us to suspect that the bottleneck lies in the network, but we have no conclusive data to support this.

7. Conclusions

There seems to be common agreement that Web growth is exponential, but there has been no quantitative data indicating the magnitude of the exponent, nor the factors that cause this growth. Through server log analysis of a variety of sites, we have determined that site growth (in terms of number of hits) correlates with one of six different quantities: the number of Web users, the number of documents a user is likely to visit on a site, the number of documents on a site, the fee structure for accessing data, the frequency with which search engines return a particular site, and the efforts of Web masters at attracting users. In addition, we have dispelled certain widely held perceptions: that CGI is becoming increasingly important in general and that heavily loaded servers are the main cause of Web latency. Finally, we quantified the effects that key design parameters have in maximizing the resource utilization of persistent connections. There remains much work to be done. In particular, detailed analysis of some of the most heavily accessed sites on the Web would be generally useful to the research community. And, while we have ruled out certain causes for latency, the answer to the question, “Why do users wait on the Web?” still eludes the research community.

8. Bibliography

[1] Bestavros, A., “WWW Traffic Reduction and Load Balancing Through Server-Based Caching,” *IEEE Concurrency: Special Issue on Parallel and Distributed Technology*, vol. 5, pp. 56-67, Jan-Mar 1997.

[2] Bowman, C., Danzig, P., Hardy, D., Manber, U., Schwartz, M., The Harvest Information Discovery and Access System. Computer Networks and ISDN Systems 28 (1995) pp. 119-125.

[3] Edwards, N., Rees, O. “Performance of HTTP and CGI,” Available at `http://www.ansa.co.uk/ANSA/ISF/1506/APM1506.html`

[4] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T., Hypertext Transfer Protocol—HTTP/1.1. *Internet Engineering Task Force Working Draft*, August 1996.

[5] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Berners-Lee, T., Hypertext Transfer Protocol—HTTP/1.1, RFC-2068, ftp://ds.internic.net/rfc/rfc2068.txt.

[6] Gwertzman, J., Seltzer, M., “The Case for Geographical Push-Caching,” Proceedings of the Fifth Workshop on Hot Topics in Operating Systems, Orcas Island, WA, May, 1995, 51–55.

[7] Manley, S., “An Analysis of Issues Facing World Wide Web Servers,” Harvard University, Computer Research Laboratory Technical Report, TR-12-97, July 1997.

[8] Mogul, J., “Network Behavior of a Busy Web Server and its Clients,” Digital Equipment Corporation Western Research Lab Technical Report DEC WRL RR 95.5.

[9] Mogul, J., “The Case for Persistent Connection HTTP,” *Proceedings of the 1995 SIGCOMM ‘95 Conference on Communications Architectures and Protocols*.

[10] Nielsen, H., Gettys, J., Baird-Smith, A., Prud’hommeaux, E., Lie, H., Lilley, C., Network Performance Effects of HTTP/1.1, CSS1, and PNG. W3 Consortium Note available at `http://www.w3.org/pub/WWW/Protocols/HTTP/Performance/Pipeline.html`