In *World Wide Web,*
Special Issue on Characterization and Performance Evaluation

Changes in Web Client Access Patterns

Characteristics and Caching Implications

Paul Barford   Azer Bestavros   Adam Bradley   Mark Crovella

Computer Science Department

111 Cummington St.

Boston University

Boston, MA 02215

Abstract

Understanding the nature of the workloads and system demands created by users of the World Wide Web is crucial to properly designing and provisioning Web services. Previous measurements of Web client workloads have been shown to exhibit a number of characteristic features; however, it is not clear how those features may be changing with time. In this study we compare two measurements of Web client workloads separated in time by three years, both captured from the same computing facility at Boston University. The older dataset, obtained in 1995, is well-known in the research literature and has been the basis for a wide variety of studies. The newer dataset was captured in 1998 and is comparable in size to the older dataset. The new dataset has the drawback that the collection of users measured may no longer be representative of general Web users; however using it has the advantage that many comparisons can be drawn more clearly than would be possible using a new, different source of measurement. Our results fall into two categories. First we compare the statistical and distributional properties of Web requests across the two datasets. This serves to reinforce and deepen our understanding of the characteristic statistical properties of Web client requests. We find that the kinds of distributions that best describe document sizes have not changed between 1995 and 1998, although specific values of the distributional parameters are different. Second, we explore the question of how the observed differences in the properties of Web client requests, particularly the popularity and temporal locality properties, affect the potential for Web file caching in the network. We find that for the computing facility represented by our traces between 1995 and 1998, (1) the benefits of using size-based caching policies have diminished; and (2) the potential for caching requested files in the network has declined.

1 INTRODUCTION

Understanding the nature of the workloads and system demands created by users of the World Wide Web is crucial to properly designing and provisioning Web services. While much can be inferred from observing the request streams arriving at servers, a complete picture of Web workloads requires measurements of the behavior of clients.

Previous measurements of Web client workloads have been shown to exhibit a number of characteristic features. In this paper we concentrate on two general categories of workload characteristics: first, statistical characterization of client requests; and second, the potential benefits from the use of caching in the network to satisfy client requests.

The statistical properties of Web client workloads are typically characterized by high variability. Previous studies have shown that the variability in file sizes, transfer times, and request interarrivals tend to be very high (see [Feldmann and Whitt 1997] for a review). Often the best distributional model for such highly variable datasets seems to be one with a *heavy tail*, that is, one whose upper tail declines like a power-law with exponent less than 2. Random variables with heavy tails have infinite variance; in practice this is exhibited as lack of convergence of the sample variance to any value, as sample sizes are increased [Crovella and Lipsky 1997].

The potential for the use of caching to satisfy requests is of considerable interest to designers of Web transfer protocols and Web infrastructure developers. The effectiveness of caching schemes relies on the presence of temporal locality in Web reference streams and on the use of appropriate cache management policies that are appropriate for Web workloads. Much previous work has focused on characterizing reference locality in Web reference streams, and typical results have shown that the potential benefits of caching in the network (or proxy caching) are moderate, typically in the 30-50% range [Bestavros *et al.* 1995; Abrams *et al.* 1995].

While these properties have been well documented in the literature, an important question concerns how these properties may be changing over time. For example, these properties were all noted in data collected during early 1995 at Boston University and discussed in [Cunha *et al.* 1995; Bestavros *et al.* 1995]. However the Web and the uses to which it is put have changed enormously since 1995. In many ways, the Web in early 1995 was in a nascent state, not yet supporting the sophisticated information sources and applications that are crucial components of the Web today. Just as important is the fact that today's users of the Web are quite different from those in 1995: a much wider population segment is familiar with the Web and uses it regularly; and the users of the Web may take for granted a wider variety of information sources and media types than did the users of 1995.

As a result, it is important to ask whether the observed statistical and caching properties of Web workloads are changing over time, and in particular whether they have changed since 1995. To answer this question we undertook new measurements of Web client workloads, in the same computing facility used for the 1995 study. Our measurements span a period of seven weeks from April 4 to May 22, 1998 and the resulting dataset is comparable in some size respects to the older dataset. Both datasets were collected in a laboratory of Sun SparcStation 2 workstations used primarily by undergraduates in the Boston University Computer Science Department. While the 1995 data was collected using an instrumented version of NCSA Mosaic, the 1998 data was gathered using non-caching HTTP proxy software which recorded all requests made by uninstrumented Netscape Navigator browsers.

While it may have been reasonable to consider the users in our laboratory in 1995 typical Web users, it is not clear that the same is true for our 1998 study. Significant Web content is now only accessible on personal computer platforms such as Windows 95. However, the value in this study is precisely that by focusing on a single computing environment, we have controlled for a number of factors that otherwise would not be controlled. In particular, the type of work being performed by Web users, and in the general purposes for which the Web is being used, are the same in both traces.

Using these two datasets, we conducted statistical analyses of the 1998 data and formed comparisons between the old and new datasets both in their statistical and cacheability properties. Of particular interest were questions of the distributions of requested and unique document sizes, the popularity of documents relative to each other and as a function of their size, as well as trace-driven studies of cacheability.

Our statistical observations serve to reinforce and deepen our understanding of the characteristic statistical properties of Web client requests. With respect to the distributional properties of Web requests, we find that while model parameters have changed, that the kinds of distributions that are appropriate to model workloads have not changed. File size distributions are still well modeled as hybrids having lognormal bodies, and power-law (*i.e.,* Pareto) tails. This suggests that the hybrid lognormal-Pareto model is still a valid one for characterizing file sizes in the Web. We also find that the characteristic nature of file popularity, in which a small set of documents receives the majority of requests, is present in both datasets, but is less pronounced in the 1998 dataset than in the 1995 dataset.

The suggestion that popularity profiles may be less extreme in the newer dataset suggests the need to compare the effects of caching applied to the two workloads. We are particularly concerned with the potential benefits of file caching in the network, that is, between the client and the server. We show that in evaluating cache replacement policies, quite different conclusions are obtained depending on whether one is concerned with the hit rate (fraction of requests that hit in the cache) or the byte hit rate (fraction of bytes that are served by the cache). We show that the difference between these two metrics is a direct function of the covariance of the number of requests per file and the sizes of files. For example, our conclusions examine the covariance between document popularity and document size, and show that for the same type of users, in the same computing environment, this value has changed significantly (by a factor of 20). Our results also indicate that for the computing facility represented by our traces, the potential for caching requested files in the network has declined from 1995 to 1998. We explore several hypotheses regarding this difference, including increased effectiveness of the browser's cache, the degree of temporal overlap of user sessions, and the degree of content overlap of user sessions.

2 RELATED WORK

The statistical properties of Web workloads that we study in this paper have also been examined in a number of other studies. One of the first studies to quantitatively explore user behavior was [Catledge and Pitkow 1995]. A detailed look at the properties of server workloads is presented in [Arlitt and Williamson 1997]. Subsequent significant studies include [Abdulla *et al.* 1997; Manley and Seltzer 1997; Deng 1996; Iyengar *et al.* 1998]. Finally, our previous work has been reported in [Cunha *et al.* 1995; Bestavros *et al.* 1995; Almeida *et al.* 1996; Crovella *et al.* 1998; Crovella and Bestavros 1997; Barford and Crovella 1998]. A good recent review of progress in characterizing Web workloads is given in [Pitkow 1997].

In characterizing the relative number of requests made to different Web documents, previous work has often referred to *Zipf's law* [Zipf 1949, discussed in [Mandelbrot 1983]]. Zipf's law was originally applied to the relationship between a word's popularity in terms of rank and its frequency of use. It states that if one ranks the popularity of words used in a given text (denoted by ρ) by their frequency of use (denoted by P) then

$$P \sim \rho^{-\beta}$$

with β typically near 1. More generally, Zipf-like laws relate frequency of symbol use to popularity rank via a power-law relationship.

Zipf-type distributions of document popularity were first noted in Web data in [Glassman 1994]. Since then the presence of Zipf-type distributions in the Web has been noted in a number of other studies, among which are [Cunha *et al.* 1995; Almeida *et al.* 1996; Nishikawa *et al.* 1998]. A recent attempt to explain how Zipf-like laws arise in the context of Web use appears in [Huberman *et al.* 1998].

Work on the interaction of Web workloads with caches has also been extensive. In [Glassman 1994], Glassman presents one of the earliest attempts for caching on the Web, whereby proxy caches are organized into a tree-structured hierarchy with cache misses in lower relays percolating up through higher relays until the requested object is found. The performance of this caching system for a single relay with a rather small cache size indicated that it is possible to maintain a *"fairly stable"* 33% hit rate. Using a Zipf-based model, it was estimated that the maximum achievable hit rate is 40%. A similar result is described in [Abrams *et al.* 1995], where the authors found that caching proxies have a maximum possible hit rate of between 30 and 50%, and in [Maltzahn *et al.* 1997], which notes a caching proxy hit rate of about 30%. The results we present in Section 5 of this paper agree with these conclusions.

The results in [Abrams *et al.* 1995] also focused attention on cache replacement policies other than Least-Recently-Used (LRU); additional replacement policies have been proposed in [Williams *et al.* 1996], [Bolot *et al.* 1997; Bolot and Hoschka 1996], [Murta *et al.* 1998], and [Markatos 1996]. As a result, we consider the Largest-File-First (LFF) policy in our cache simulations as well as the LRU policy. While [Abrams *et al.* 1995] proposed policies that give preference to small files over large files, we find in this paper that the relative benefit of such policies has declined in our environment from 1995 to 1998.

Thus, while previous studies have addressed a wide and deep range of questions, none specifically looked at how workloads change over time in any particular environment—which is the focus of this paper.

3 DATA COLLECTION

Our study is driven by two sets of client traces, one from 1995 and one from 1998. Traces consist of records of Web objects ("files" or "documents," which terms we use interchangeably) requested by users and usually transferred over the Internet. As discussed in [Cunha *et al.* 1995], the 1995 trace was collected from November 1994 through February 1995 using an instrumented version of Mosaic, the GUI web browser of choice at the time. This approach allowed the collection of prolific data about the particulars of user's browsing habits, including clearly demarcated *sessions* (a single execution of the browser), a record of all *requests* for documents (including those served out of Mosaic's document cache), the *relationship* of documents to each other (which ones were requested by a user action as opposed to those requested "implicitly", e.g. inline images), *timestamps* for all events, and *durations* of all file transfers over the network. Traces were recorded for a total of 591 users on 37 machines.

During the course of the 1995 study, Netscape's Navigator web browser became prominent in our departmental computer labs, and it remains so to this day. Since Navigator's source code was not available when we decided to undertake this new study,[1] it was impractical to consider using a modified browser to collect data as we had in 1995. Instead, it was decided that lightweight non-caching HTTP proxies would be used to track all document references made by the unmodified Navigator clients run by the majority of users in our lab. The proxies collected traces for 306 users on 29 machines.

Note that the workstations were not in use continuously during either of the measurement periods. Our measurements show that for the 1998 study, that the average rate that HTTP requests were made while

[1] The first Mozilla 5.0 pre-beta source code release was on March 31, 1998, just a few days prior to the beginning of our data collection.

a workstation was actually in use for browsing was about 2 requests per minute.

In the remainder of the paper we will refer to the dataset collected in 1995 as the **W95** dataset and the dataset collected in 1998 as the **W98** dataset.

3.1 Trace Strategy

The proxy server used was an adaptation of a simple, multi-threaded web server developed as part of a related project in our department. The HTTP proxy functionality was implemented strictly as a data-collection tool, with no attempts at caching, connection aggregation, or other optimizations.

This proxy server was installed on *each* of the workstations in a laboratory used primarily by undergraduates in Boston University's Computer Science Department. We then replaced the Communicator binaries on our lab machines with a shell script which would start Communicator with its "Proxy Auto-Configuration" [2] feature enabled. A simple CGI script on our departmental web server provided the auto-configuration scripts to browsers based upon their hostname, allowing us the flexibility (in case it had been necessary) to migrate to a central proxy or disable the proxy for a subset of the workstations without significantly interfering with data collection or with the lab's working environment.

3.2 Log Format

The proxy server on each workstation maintained its own append-only text log file on a shared NFS filesystem. These log files were later combined to produce the final trace. The proxy server recorded the following information about each request it processed: the method of the request (GET, POST, etc); the web server name; the URL; the protocol used by the requesting agent (all HTTP/1.0 in this trace); the client's IP address and port number used to connect with the proxy; the proxy's IP address and port number used to connect with the upstream server; the server's IP address and port number used to serve the request; the name of the client's user as reported by the RFC1413 *ident* service[3]; the status code returned by the server (usually 200); the length of the content returned by the server (or -1 if a transport or protocol error occurred); a series of timestamps (seconds and microseconds since 0:00:00 1/1/70 GMT); the "Referrer" field as reported by the client; and the User-Agent field as reported by the client.

[2] `http://www.netscape.com/eng/mozilla/2.0/relnotes/demo/proxy-live.html`

[3] this worked very well in our environment, since all client machines were running identical servers. All failures were caused by clients closing their connections prematurely, i.e. pressing their "Stop" buttons or crashing.

3.3 Comparison of Proxy Traces with Browser Traces

A distinguishing feature of our study is that we are able to look at two client trace sets taken from the same computing facility, separated widely in time. The computing facility is a general-purpose workstation lab used primarily by undergraduate majors in Computer Science, both in 1995 and in 1998. The traces were taken during approximately the same portion of the academic year.

There are also differences in data collection methods between 1995 and 1998. The principal difference concerns having only the "network view" of web utilization visible to an HTTP proxy. In collecting the 1998 trace we did not have the ability to record the internal state (or activity) of the browser, and are missing in particular three kinds of useful information: session start and end times in the sense used in the 1995 trace, relationships of multiple requested documents (clicked-to vs. embedded documents), and documents viewed out of the browser's cache. The first two can be approximated using various time heuristics and the "Referrer" field recorded by the proxy; the impact of the third upon our study is discussed in sections 4.5 and 5.

There are several less critical drawbacks as well. A proxy server, no matter how thin, introduces some latency and processing time to request service, so any metrics derived from precise time measurements should be taken with a grain of salt. Also, since the proxy only implemented HTTP (no FTP or Gopher traffic was recorded), non-HTTP documents were removed from the 1995 trace for comparison purposes. These documents accounted for roughly 4% of the requests in the 1995 trace.

4 STATISTICAL CHARACTERIZATION

There are a wide range of statistical distributions important to Web client characterization. In this section we focus on only two of them: the size distribution of the files successfully transferred over the network, and the size distribution of the set of unique files. The set of unique files is the subset of the transferred files in which each individual file appears only once.

We studied the set of transferred files because this set influences the properties of network traffic; and we studied the set of unique files because this set can give insight into the set of files available on Web servers (as discussed in [Crovella *et al.* 1998]). In a few cases we will also look at the set of requested files in the W95 dataset, to help understand other differences between the W95 and W98 datasets. The set of requested files consists of those files requested by users, some of which were served from Mosaic's local cache, and the rest of which constitute the set of W95 transferred files.

4.1 Statistical Properties

An important question is whether these two datasets continue to show the heavy-tailed property in 1998 that was first identified in 1995. This property is important because it has been suggested as a causal mechanism for the presence of *self-similarity* (burstiness at a wide range of timescales) in Web traffic [Crovella and Bestavros 1997].

For a random variable X with distribution function F we say that F is heavy tailed if

$$P[X > x] \sim x^{-\alpha}, \quad \text{as } x \to \infty, \;\; 0 < \alpha < 2 \tag{1}$$

where $f(x) \sim y$ means that $lim_{x\to\infty} f(x)/y = c$ for some constant c. The simplest heavy-tailed distribution is the Pareto distribution with $F = 1 - (k/x)^{\alpha}, \;\; x \geq k$.

Initial statistical analysis of unique file sizes and transferred files from the 1995 BU client data focused on the tails of those distributions, as reported in [Crovella *et al.* 1998]. However, the models for the tails of those distributions were not good fits for the bodies. This lead to the development of *hybrid* models for each distribution, as reported in [Barford and Crovella 1998]. The hybrid models for the 1995 BU client data typically combined a Pareto distribution for the upper tail with lognormal distributions for each of the bodies. A lognormal distribution is one in which the logarithm of the random variable follows a normal distribution. This distribution also has a long upper tail and can be used to model highly variable data.

4.2 Methods Used

In developing models to fit file sizes distributions we followed the same analytical steps described in [Barford and Crovella 1998]. In this section we describe those steps in a general manner to organize the subsequent discussion; the specific tests performed are described in the next sections. The steps we performed are:

1. First, we used log-log complementary distribution (LLCD) plots to visually inspect whether or not the data set has a heavy tail. An LLCD plot graphs $\log \bar{F}(x) = \log(1 - F(x))$ versus $\log(x)$ for large x. A random variable with a heavy-tailed distribution will exhibit a straight line on such a plot (as is clear from Equation 1), with the line's slope an estimate of the α parameter of the distribution.

2. Next, we used standard visual techniques—simple histograms or cumulative distribution function (CDF) plots—to narrow the set of candidate models for the body of the distribution. Logarithmic transformation may be helpful to distinguish important characteristics when datasets show long tails.

Statistic	W95	W98
Sample Size	54,438	41,049
Minimum	3	1
Maximum	20,135,435	4,092,928
Mean	27,086	7,609
Median	2,833	2,769
Standard Deviation	240,237	33,306

Table 1: Simple statistical comparison of unique file size data sets (file sizes in are bytes).

3. If the data appears to be well modeled by a hybrid model (distribution for body differs from that of tail) then we used censoring methods to determine how to divide the body from the tail of the data.

4. To estimate parameters for candidate models for the data, we used maximum likelihood estimators.

5. Next, we used a goodness of fit *test* (the Anderson-Darling (A^2) [D'Agostino and Stephens 1986] test) to see if there is a close fit between model and data. If this test showed no significance, then we used random subsampling to test goodness of fit for smaller sample sizes.

6. Finally, we used a goodness of fit *metric* (the λ^2 [Pederson and Johnson 1990] metric) to determine a measure of discrepancy between data and model.

4.3 Unique Files

The unique file size model developed for the W95 dataset for the study in [Barford and Crovella 1998] consisted of a hybrid lognormal-Pareto. A lognormal distribution was used to model the body of the data while a Pareto distribution was used to model the tail.

The simple statistical comparison between the unique files in W95 and W98 is given in Table 1. These statistics indicate that the size of unique files is somewhat smaller in W98 than in W95. We note however that while the medians differ by only 2 percent, the means are radically different. This is because the empirical mean can be influenced by a few large observations, and so the heavy-tailed distribution of file sizes makes the empirical mean an extremely unstable metric (see [Crovella and Lipsky 1997]). As a result, drawing conclusions about typical file sizes based on the mean is difficult, and it is much more informative to consider the entire distribution of sizes.

The LLCD for the unique files in the W95 and W98 datasets is shown is Figure 1. This plot indicates that both datasets seem to exhibit heavy tails, which indicates that a hybrid model consisting of distribution

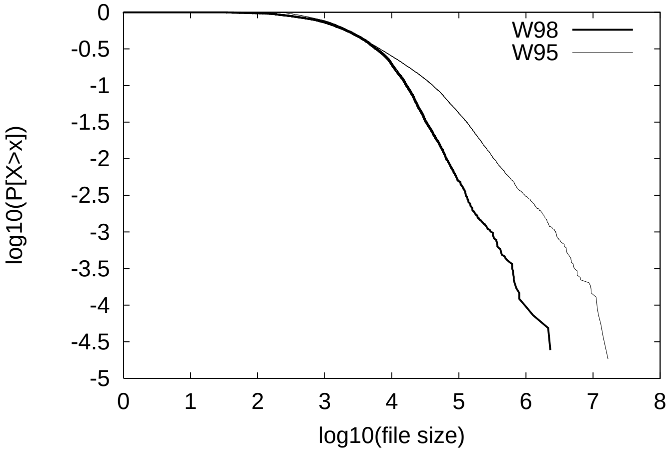


Figure 1: LLCD of unique files from W95 vs. W98.

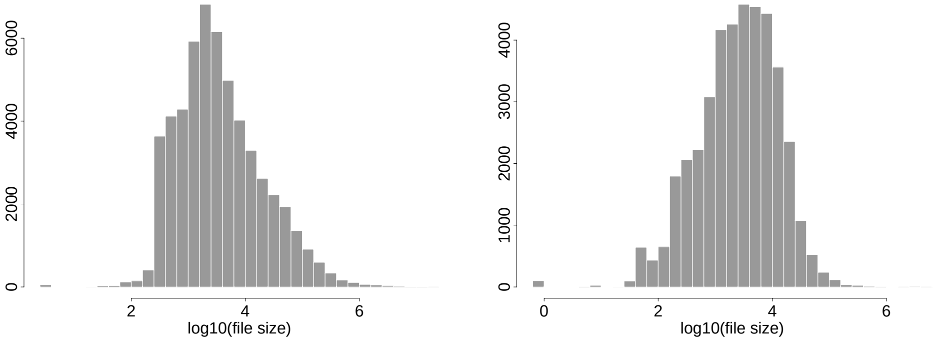


Figure 2: Histograms of log-transformed unique file sizes for W95 (left) and W98 (right).

with a heavy tail will be necessary to model the W98 dataset. It is also clear from the plot that the set of unique files in the W98 dataset is *less* heavy-tailed than in the W95 dataset, meaning that size variability is less pronounced in the W98 dataset.

In order to determine what model might be appropriate for the body of the distribution of unique file sizes in W98, we analyzed the distribution using a histogram and CDF plots. The histograms for log transforms of unique file sizes in W95 and W98 are shown in Figure 2. The CDF of the log transformed unique files sizes from W95 and W98 is shown in Figure 3. In this figure, the W98 dataset is the upper line over the entire plot. That is, it has a heavier tail on the left, and a lighter tail on the right. Thus the W98 dataset contains slightly more small files, and slightly less large files, than the W95 dataset. These plots suggest—and goodness-of-fit metrics confirm—that the lognormal distribution is a good model for the body of the unique file size distribution.

Determining where to break between body and tail can be done via censoring methods. These methods

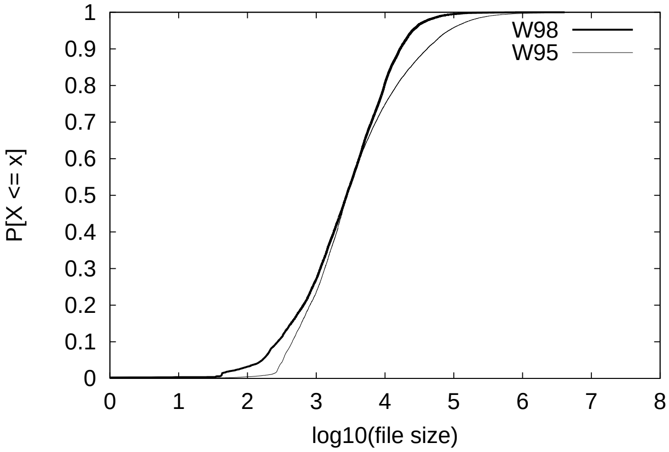


Figure 3: CDF of Log-transformed unique file sizes for W95 vs. W98.

Component	Model	W95	W98
Body	Lognormal	$\mu = 9.357$, $\sigma = 1.318$	$\mu = 7.796$, $\sigma = 1.625$
Tail	Pareto	$k = 9300$, $\alpha = 1.0$	$k = 3174$, $\alpha = 1.47$
Percent files in tail		7%	17%

Table 2: Model parameters for unique file size models.

indicate that the break point between the two distributions occurs when 17% of the data is assigned to the tail. This corresponds to a value of about 10KB.

We next calculated goodness of fit for the body using the A^2 method. This method showed that the null hypothesis that the censored sample was from a lognormal distribution (as well as all other tested distributions) must be rejected. However, applying the A^2 method on random subsamples of size 100 did return some positive results for a good fit for the lognormal distribution. The λ^2 test showed that the best fit for the body of the data was the lognormal distribution.

Since we have determined that the tail of the distribution is heavy, we modeled it with a Pareto distribution. We used standard maximum likelihood estimator (MLE) methods to determine the α value for the Pareto distribution. The results are shown in Table 2. The table confirms that unique files tend to be somewhat smaller in the W98 dataset. This is shown in both the μ parameter of the lognormal distribution and the α parameter of the Pareto.

Using the parameters listed in Table 2 we can compare the model with the original dataset. The CDF and LLCD for the data from W98 versus the hybrid model are shown in Figure 4, which shows that the model appears to be a close fit to the data.

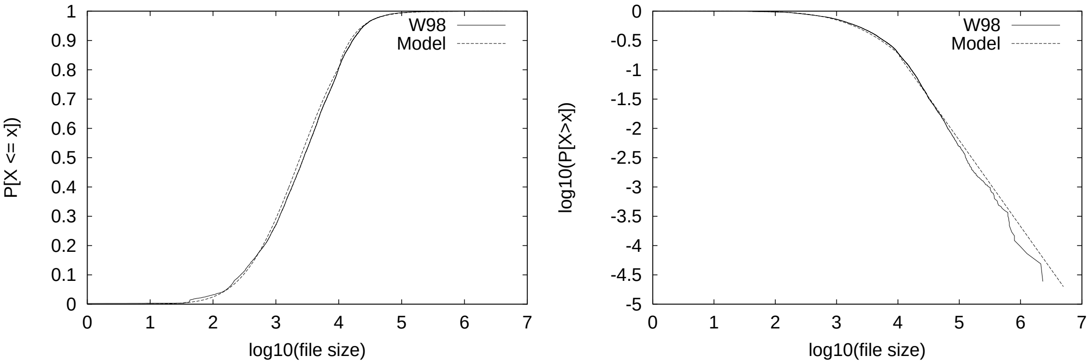


Figure 4: CDF and LLCD of log-transformed unique file sizes for W98 and the hybrid lognormal-Pareto model for W98.

Statistic	W95	W98
Sample Size	269,811	66,988
Minimum	3	1
Maximum	20,135,435	4,092,928
Mean	14,826	7,247
Median	2,245	2,416
Standard Deviation	137,399	28,765

Table 3: Simple statistical comparison of transferred file size data sets.

4.4 Transferred Files

The transferred file size model developed for the data from W95 for the study in [Barford and Crovella 1998] also consisted of a hybrid lognormal-Pareto. A lognormal distribution was used to model the body of the data while a Pareto distribution was used to model the tail.

The simple statistical comparison between the set of transferred files in W95 and W98 is given in Table 3. The table shows that there is again a relatively small difference in median file size, while the other metrics (mean and standard deviation) show less stability.

The LLCD for the set of transferred files in W95 and W98 is shown is Figure 5. This plot indicates that both datasets appear to exhibit heavy tails. It is also clear from the plot that—like the unique file size data presented earlier—the data from W98 is *less* heavy-tailed than from W95; however the difference between the two slopes is much less pronounced than in the case of unique files.

The histograms for the log transforms of transferred files in W95 and W98 are shown in Figure 6. The

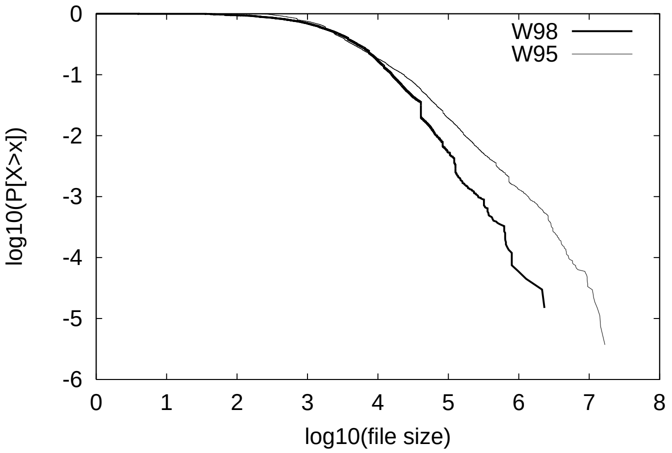


Figure 5: LLCD of transferred file sizes from W95 vs. W98.

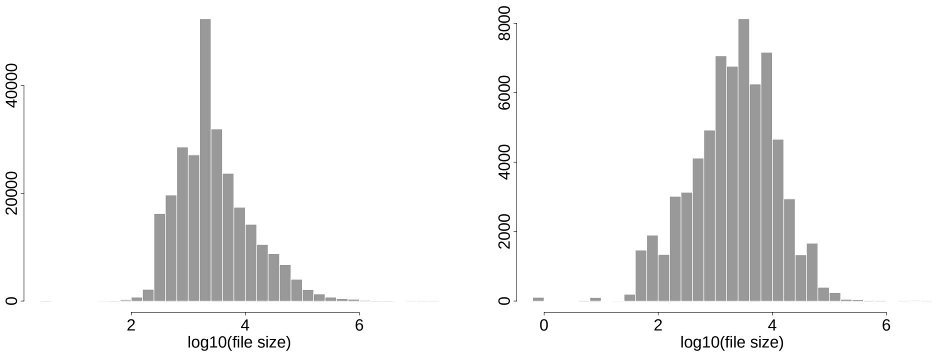


Figure 6: Histograms of log-transformed transferred file sizes in W95 (left) and W98 (right).

CDF of the log transformed transferred files sizes from W95 and W98 is shown in Figure 7. Using censoring methods, the break point between body and tail was found to be when 12% of the data was assigned to the tail (corresponding to a cutoff value of about 13.5KB). As with the set of unique files, applying the A^2 method on random subsamples of size 100 did return positive results for goodness of fit for the lognormal distribution, and the λ^2 test showed that it was the best fit for the body of the data.

The CDF and LLCD for the data from W98 compared to the hybrid model are shown in Figure 8. These figures show that the model is a good fit for the data.

These distributional results show that in the case of Web transfers, differences between the W95 and W98 datasets are much less extreme. The two datasets are much closer in tail weight and in the mean of the lognormal distribution.

In summary, we find that distributionally, there are not great differences between the W95 and W98 datasets. The same kinds of distributional models are appropriate for both datasets, with only the

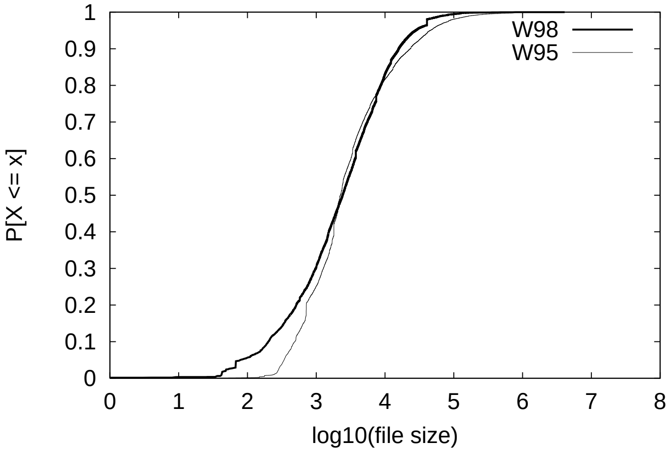


Figure 7: CDF of Log-transformed transferred file sizes in W95 vs. W98.

Component	Model	W95 Parameters	W98 Parameters
Body	Lognormal	$\mu = 7.881$; $\sigma = 1.339$	$\mu = 7.640$; $\sigma = 1.705$
Tail	Pareto	$k = 3,558$; $\alpha = 1.177$	$k = 2,924$; $\alpha = 1.383$
Percent files in tail		7%	12%

Table 4: Model parameters for transferred file size models.

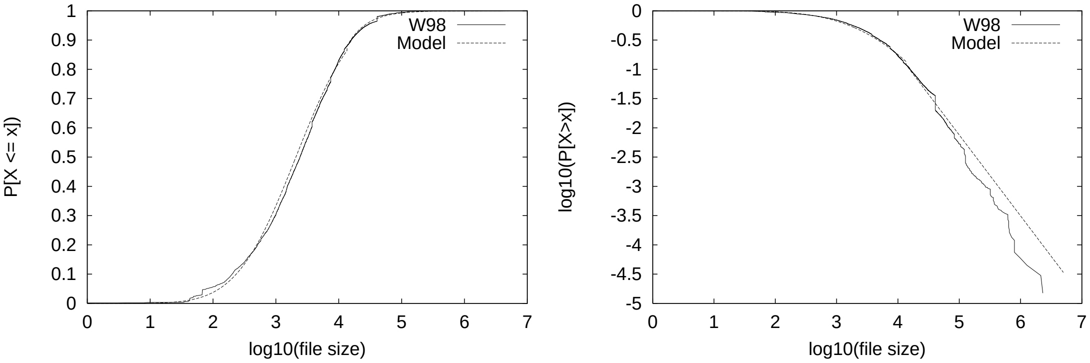


Figure 8: CDF and LLCD of log-transformed transferred file sizes for W98 and corresponding hybrid lognormal-Pareto model.

distributional parameters changing between W95 and W98. This indicates that these aspects of the workloads carried by networks due to the Web may not have changed radically over the three-year timespan from 1995 to 1998

4.5 Relative Document Popularity

The last statistical property we examine concerns Zipf's Law, which (as described in Section 2) can characterize the relative popularity of documents in the Web. In this section we confirm that Zipf laws appear in all of our datasets. More importantly however, we also use Zipf law plots to study differences in the relative popularity of documents between the 1995 and 1998 datasets.

As discussed in [Cunha *et al.* 1995], the set of requests in the 1995 dataset strongly exhibits Zipf's law. This effect is shown in Figure 9. The upper line in the figure shows the log-transformed plot of the number of references to each document as a function of its rank in popularity. Along with the points is the least-squares fit line ($R^2 = 0.99$) showing a slope of -0.96. Thus for the set of requested documents in 1995, $\beta = 0.96$.

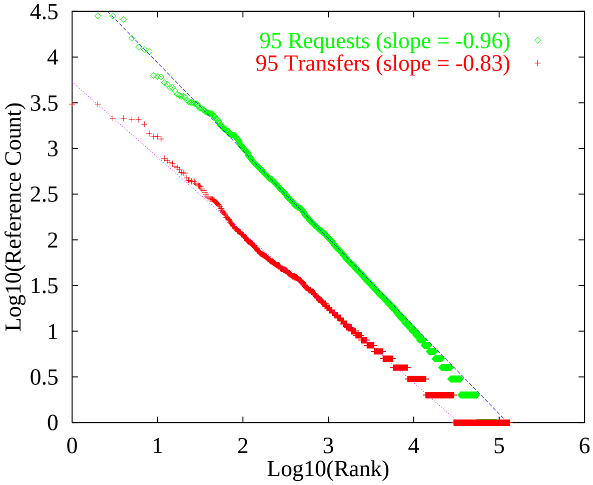


Figure 9: Zipf laws in 1995 Requests and Transfers.

Also shown in the figure is the same plot for the set of transfers in 1995, along with its least-squares fit line ($R^2 = 0.99$). This plot shows a distinctly different slope, confirmed by the fitted line which yields $\beta = 0.83$.

The difference between these two datasets arises because the set of transfers is the set of cache misses resulting from the set of requests. Documents that tend to hit in the cache will be those that are requested most often. Thus, when comparing the set of requests with the set of cache misses, one would expect that popular documents would be preferentially removed by the action of client caches. As a result, the Zipf law shows a smaller β for the set of transfers as compared to the set of requests.

Surprisingly, a smaller β is also seen when comparing the set of transfers in 1995 with the set of transfers in 1998. This comparison is shown in Figure 10. The figure shows that for the 1995 transfers, $\beta = 0.83$ (as before) but that for the 1998 transfers, $\beta = 0.65$ ($R^2 = .99$).

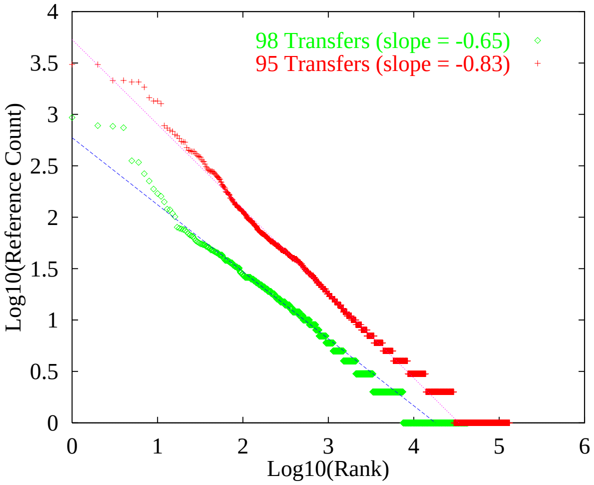


Figure 10: Zipf laws in 1995 Transfers and 1998 Transfers.

This difference shows that, relatively speaking, the most popular documents are less popular in the 1998 transfer dataset than in the 1995 transfer dataset. That is, references to documents in the 1998 dataset are spread more evenly among the set of documents.

By analogy with the previous comparison (1995 requests vs. 1995 transfers), it is possible that this effect is caused by improved caching at the client in 1998. That is, we speculate that more effective caches at the client would tend to reduce the repeated requests for popular files. A possible consequence of this would be that the performance of caches in the network (downstream from the client) is reduced in the 1998 dataset as compared to the 1995 dataset. While the Zipf plots do not provide final proof for either of these conclusions, they suggest that the effect of network caching may be quite different between the 1995 and 1998 datasets; in the next section we explore this question in more detail.

5 NETWORK CACHING IMPLICATIONS

Web caching opportunities exist at a multitude of points between a client and a server. Caches may exist within the client software (browser cache), within the network (network cache), or at the server (server "accelerators" or front-ends). The effectiveness of a cache at any of these points depends on the temporal locality present in the reference stream at that point. A major contributor to such referential locality is the well-documented Zipf laws governing popularity of Web documents—namely, that a large portion of the requests is for a small subset of documents. Given the changes we observed in document popularity profiles (*i.e.,* Zipf plots) between 1995 and 1998 (see Figure 10 in Section 4), the question arises: what are the implications of these changes on the effectiveness of network caches? This section attempts to answer this question through a set of trace-driven simulation experiments.

To conduct our experiments, a simple HTTP cache simulator was implemented. The simulator takes as parameters the cache size and a replacement algorithm; and as input a stream of document URLs and their respective sizes. The simulator checks for hits using a simple URL match,[4)] and performs on-demand document eviction. The simulator calculates and reports the two metrics of interest to us, *Hit rate (H)* and *Byte hit rate (B)*. Hit rate is the proportion of requested *documents* that are served out of the cache; Byte hit rate (also called "Weighted hit rate" in [Pitkow 1997]) is the proportion of requested *bytes* that are served out of the cache.

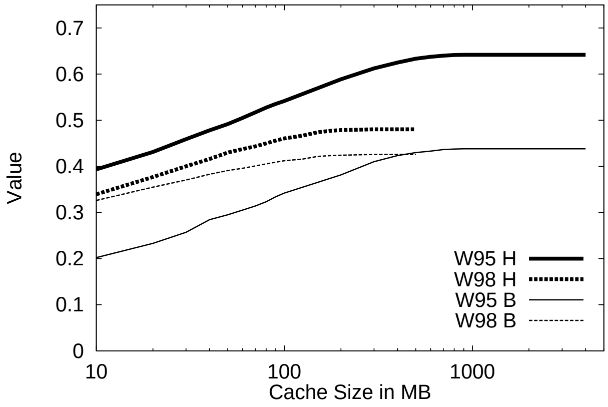


Figure 11: LRU document hit rate (H) and byte hit rate (B) as a function of network cache size.

5.1 Baseline Trace Simulations

Figure 11 shows the performance of a network cache utilizing the Least Recently Used (LRU) replacement policy. The two metrics H and B are plotted as functions of the network cache size.[5)]

At first glance, Figure 11 indicates that the hit rates (H) for W98 are *significantly lower* than the hit rates for W95. In particular, our simulations indicate that the value of H levels at around 64% for W95

[4)] Neither W95 nor W98 contain adequate information to unambiguously determine when documents are unchanged, in order to correctly simulate an HTTP/1.1 cache. Document size is a candidate for a second match heuristic, but only the W98 trace possesses adequate data to correctly drive such a state machine. Since our focus is on using cachability to compare the two traces rather than focusing on absolute values of the metrics, the presented approach was deemed acceptable.

[5)] The 1998 simulation plots throughout this paper extend only part way across the various graphs, reflecting the smaller set of unique files in the 1998 dataset.

versus 48% for W98. This large difference all but disappears when we consider the byte hit rate metric (B), which levels at around 44% for W95 and 43% for W98. More importantly, the byte hit rate (B) for W98 is noticeably better than that of W95 when considering smaller cache sizes.

5.1.1 A Weakening Preference for Smaller Documents

The difference between the hit rate and the byte hit rate for a given trace when applied to an infinite cache is a measure of the tendency for small files to be requested preferentially over large files.[6] This relationship is quantified in the following Lemma.

Lemma 1. For some sequence of document requests let the number of unique documents present in the trace be N, and let $i, i = 1, 2, ..., N$ be an indexing of the set of unique documents present in the sequence. Let R_i be the number of requests to document i in the sequence and let S_i be the size of document i in bytes. Also define the average number of requests per document $\mu_r = \sum_{i=1}^{N} R_i/N$, the average size of a document $\mu_s = \sum_{i=1}^{N} S_i/N$, and the average number of bytes requested per document $\mu_b = \sum_{i=1}^{N} R_i S_i/N$. Let H_∞ denote the hit rate of the sequence of requests when applied to an infinite cache, and let B_∞ denote the byte hit rate of the sequence when applied to an infinite cache. Then:

$$H_\infty - B_\infty = \frac{-\mathrm{Cov}(R_i, S_i)}{\mu_r \mu_b} \tag{2}$$

where $\mathrm{Cov}(a_i, b_i)$ is the sample covariance of the sequences a_i and b_i.

Proof:

In an infinite cache, each unique document would miss exactly once. Therefore

$$H_\infty = \frac{\sum_{i=1}^{N}(R_i - 1)}{\sum_{i=1}^{N} R_i} = 1 - \frac{N}{\sum_{i=1}^{N} R_i}$$

and

$$B_\infty = \frac{\sum_{i=1}^{N}(R_i - 1)S_i}{\sum_{i=1}^{N} R_i S_i} = 1 - \frac{N\mu_s}{\sum_{i=1}^{N} R_i S_i}$$

So:

$$\begin{aligned} H_\infty - B_\infty &= 1 - \frac{N}{\sum_{i=1}^{N} R_i} - \left(1 - \frac{N\mu_s}{\sum_{i=1}^{N} R_i S_i}\right) \\ &= \frac{N\mu_s \sum_{i=1}^{N} R_i - N\sum_{i=1}^{N} R_i S_i}{\sum_{i=1}^{N} R_i \sum_{i=1}^{N} R_i S_i} \\ &= \frac{N\sum_{i=1}^{N} R_i(\mu_s - S_i)}{\sum_{i=1}^{N} R_i \sum_{i=1}^{N} R_i S_i} \\ &= \frac{\sum_{i=1}^{N} R_i(\mu_s - S_i)}{N\mu_r\mu_b}. \end{aligned} \tag{3}$$

[6] An infinite cache is one that is so large that no file in the given trace, once brought into the cache, need ever be evicted.

Now, note that

$$\sum_{i=1}^{N}(\mu_s - S_i) = 0.$$

Therefore we can rewrite Equation 3 as

$$\begin{aligned} H_\infty - B_\infty &= \frac{\sum_{i=1}^{N} R_i(\mu_s - S_i) - \sum_{i=1}^{N} \mu_r(\mu_s - S_i)}{N\mu_r\mu_b} \\ &= \frac{\sum_{i=1}^{N}(R_i - \mu_r)(\mu_s - S_i)}{N\mu_r\mu_b} \\ &= \frac{-\mathrm{Cov}(R_i, S_i)}{\mu_r\mu_b} \end{aligned} \tag{4}$$

□

Lemma 1 indicates that if there is no correlation between the size of a file and its likelihood of being accessed, then there will be no difference between hit rate H and byte hit rate B. Furthermore, a negative covariance indicates a preference for smaller files (resulting in $H_\infty > B_\infty$), whereas a positive covariance indicates a preference for larger files (resulting in $H_\infty < B_\infty$).

For the 1995 and 1998 datasets, Table 5 shows the values of the covariance between S_i and R_i ($\mathrm{Cov}(R_i, S_i)$), the mean number of requests per document (μ_r), the mean document size (μ_b), the difference between document and byte hit rates as predicted by equation 4 ($H_\infty - B_\infty$), and the difference between document and byte hit rates as observed in our simulations for the largest cache sizes ($H_{max} - B_{max}$).

For both datasets the negative covariance indicates a preference for smaller files. However, this preference is stronger in W95 than in W98.[7] Figure 12 visualizes this observation by showing the popularity of documents as a function of document sizes for W95 and W98. As one would expect from the covariance results obtained above, small documents were more popular in W95 than in W98 as indicated by the higher "request mass" for smaller document sizes in Figure 12, particularly in the 1KB-4KB range.

Thus, to summarize, the noticeable discrepancy between the values of H and B in Figure 11 is indicative of a preference for small files in the request stream. Furthermore, this preference has weakened in 1998 compared to 1995.

5.1.2 Implications for Cache Replacement Policies

What implications does this change in preference for smaller files have on the performance of the various cache replacement policies—especially those that incorporate the "size" of the requested document into the replacement decision?

[7] Note that in equation 4, the denominator for the 1998 data set is much smaller than that of 1995, so the difference in $H - B$ is due solely to the increase in covariance for the old dataset over the new dataset; the denominators differ by factor of 3.6, but numerators differ by factor of 20.

	W95	W98
Cov(R_i, S_i)	-11175.9	-608.8
μ_r	2.0548	1.6763
μ_b	18,885	7,209
$H_\infty - B_\infty$	0.2880	0.0503
$H_{max} - B_{max}$	0.2038	0.0544

Table 5: $H - B$ as an indication of preference for smaller files.

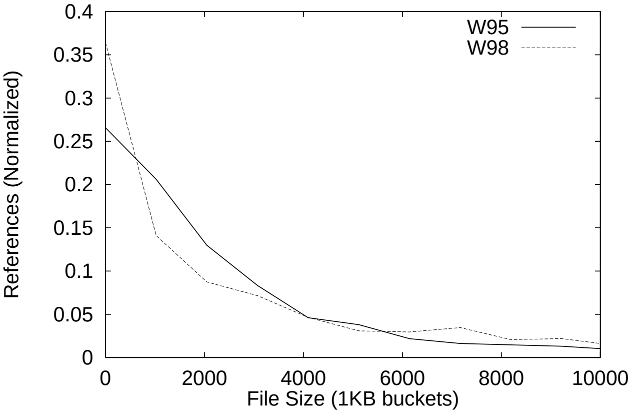


Figure 12: Number of requests as a function of document size.

To answer this question, we performed a series of trace-driven simulations to contrast the performance of three network cache replacement strategies for the W95 and W98 datasets. The three strategies we considered are Least-Recently-Used (LRU), Largest-File-First (LFF), and First-In-First-Out (FIFO). LRU and LFF are chosen as representatives of strategies that either do or do not incorporate the "size" of the requested document into the replacement decision. FIFO is chosen as a representative of strategies that do not exploit any particular access pattern characteristics, and hence its performance can be used to gauge the benefits of exploiting any such characteristics.

For each of the three strategies we measured two metrics: the document hit rate H and the byte hit rate B, for a range of network cache sizes. Figures 13(a) and 13(b) show the values of H and B, respectively, for the three strategies as a function of the network cache size.

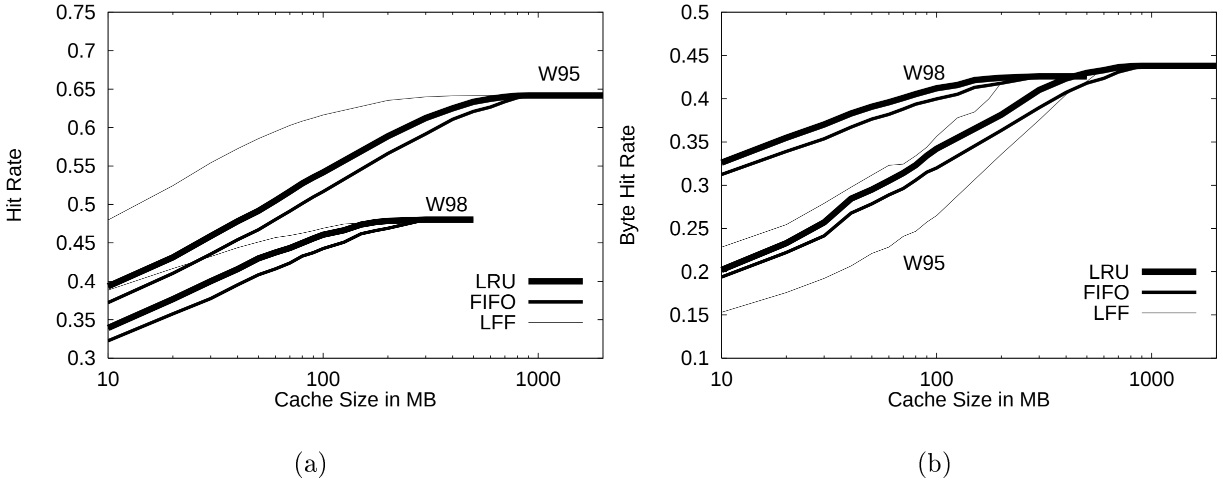


Figure 13: Performance of LRU, FIFO, LFF as a function of cache size: (a) H metric (b) B metric.

For both the W95 and W98 datasets, Figure 13 shows that LFF outperforms LRU with respect to the document hit rate metric (H), but performs significantly worse than LRU with respect to the byte hit rate metric (B). The differences are more pronounced for smaller cache sizes (*i.e.* when the replacement strategy plays a more critical role).

The superiority of LFF with respect to the H metric can be understood by noting that the goal of LFF is to pack as many documents as possible into a given fixed-size cache. It does so by giving preference to smaller documents over larger documents, without regard for the popularity of individual documents. This policy results in a higher "document density" and thus higher hit count, especially if requests are made preferentially for smaller files. This preference for smaller files is indeed present in both datasets (as shown earlier in this section), albeit weaker in W98 than in W95. In particular, the declining covariance between R_i and S_i (in W98 vs. W95) means that the differences in hit rates (H) between LRU and LFF will tend to decline as well.

The superiority of LRU with respect to the B metric can be understood by noting that the goal of LRU is to exploit the recurrence of requests in the reference stream (temporal locality of reference). It does so by giving preference to more popular documents over less popular ones (independent of size). This policy results in a higher "byte utility" in the cache, especially if locality of reference is strong. The similar performance of LRU and FIFO can be attributed to the fact that the FIFO algorithm could be said to approximate LRU assuming extremely poor locality of reference, and their relative performance could be used to gauge the potential benefits of capitalizing on such locality of reference.

Finally, comparing the performance of LRU and LFF under the two metrics yields insight into the effects of decreased $|Cov(R_i, S_i)|$ in the W98 trace. The performance improvement of LFF over LRU under the H metric is less for the W98 trace than it is for the W95 trace; furthermore the penalty paid under the B metric by using LFF rather than LRU is greater in the W98 trace than it is in the W95 trace. This experimental observation is consistent with intuition provided by the Lemma. It reflects the fact that as the absolute correlation between requests and sizes declines, as happened between 1995 and 1998 in our traces, the benefits of using LFF decline as well.

5.2 Synthetic Trace Simulations

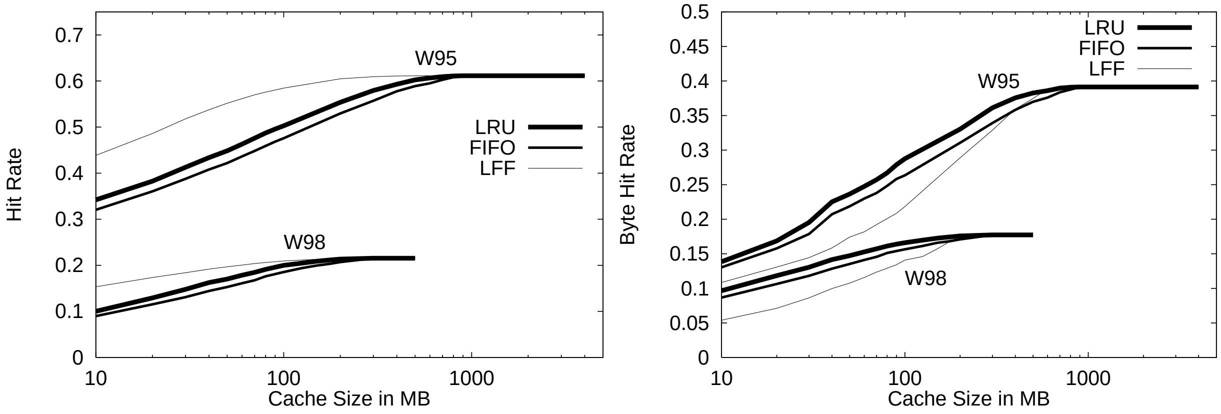
The results of the previous subsection indicate that the W98 traces showed generally poorer payoffs from caching (both in terms of H and B) than the W95 traces. We examined a number of possible cause(s) of this apparent decline in the utility of caching in the network. To do so, we used the original W95 and W98 traces to generate synthetic traces that are used in the various experiments presented in this section.

To understand the basis for the possible causes we discuss below, one needs to consider the sources of temporal locality. The temporal locality of reference in the request stream presented to a network cache can be due to recurrent requests originating within the same session (*intra-session* temporal locality) or to recurrent requests originating from different sessions (*inter-session* temporal locality). It follows that differences in temporal locality between 1998 and 1995 (and hence cache performance) could be attributed to a decline in (1) intra-session temporal locality, and/or (2) inter-session temporal locality.

5.2.1 The Effect of Client Caches

The first possible cause for the difference in network caching performance is the influence of caching in the browser. In particular, as suggested earlier, declining hit rates in the W98 trace may be due to better browser caching in that environment. Since the request stream presented to a network cache consists of exactly those requests that *missed* in the browser cache, a more effective browser cache would result in fewer misses for the same document. As a result, the request stream presented to the network cache would have less recurrent requests and hence would be less “cacheable”. This hypothesis is also advanced in [Muntz and Honeyman 1992] to explain low hit rates in secondary file caches.

While the effectiveness of the browser cache for 1995 could be evaluated (and indeed was evaluated in [Bestavros *et al.* 1995]), the same could not be done for 1998 (as explained in section 3). Thus, comparing the performance of the browser caches was not possible. Therefore, the only alternative to test this hypothesis was to “equalize” the browser caches and then evaluate the effectiveness of a network cache on the resulting streams. To do so, we assumed the existence of a “perfect client cache” for each of the datasets. A perfect client cache is one that misses each document exactly once for each user in the data set.



(a)

(b)

Figure 14: Performance of LRU, FIFO, LFF as a function of cache size under a perfect browser caching assumption: (a) H metric (b) B metric.

Figure 14 shows the hit rates and byte hit rates for the various replacement strategies under a perfect client cache assumption. With perfect client caches, the W95 data set continues to show significantly better network hit rates than the W98 data. In particular, in the presence of a perfect browser cache, the upper bound on H (B) for the W95 data set was measured at about 61% (39%), compared to 22% (18%) for the W98 data set.

Furthermore, comparing Figure 14 with Figure 13 shows that the performance improvements that can be obtained by a perfect client cache are greater for the W98 trace than for the W95 trace. That is, the network cache hit rates for W98 decreased much more than for W95 when a perfect client cache is used.

These two observations suggest that although some differences between the cache hit rates of the W95 and W98 traces may be due to improved caching in the W98 browsers: 1) the difference are not solely due to better browser caching, because even if browser caches were perfect, significant differences in network cache hit rates *still* exist;[8] and 2) it seems there are *more* opportunities still remaining for improved client caching in the W98 traces than in the W95 traces.

Thus it seems that there is less inter-session locality of reference in the W98 trace. This is suggested by the fact that if all intra-session locality of reference is removed (as in the perfect client cache traces) the resulting W95 trace shows much higher network hit rates than does the W98 trace.

5.2.2 The Effect of Multiprogramming Level

If inter-session locality of reference is an important component of network cache hit rates, then it is important to consider whether the degree to which sessions were concurrent in each trace has affected our results. If there is a significantly higher interleaving of sessions in the W95 trace this may cause higher hit rates for files that are requested in multiple sessions.

To see whether this is the case, we profiled the distribution of concurrent sessions—a.k.a. the *multiprogramming level* (MPL)—in the W95 and W98 datasets. Since explicit session start and end times were not available for the W98 trace, a per-user timeout of 30 minutes was used to separate traces into sessions. Figure 15 shows that, indeed, the W95 data set exhibited a larger MPL than that exhibited in the W98 data set. In particular, the minimum, average, and maximum MPLs were 1, 2.27, and 21, respectively for W95, versus 1, 1.73, and 11, respectively for W98.

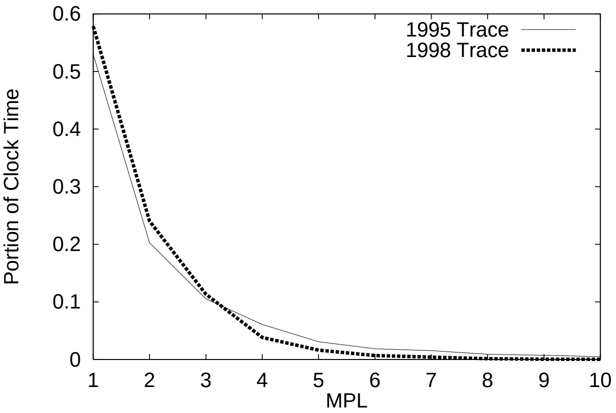


Figure 15: Characteristics of the MPL for the W95 and W98 data sets.

Figure 15 indicates that there is a difference in MPLs between W95 and W98, but it doesn't indicate whether this difference is translated into a difference in the level of inter-session temporal locality—and

[8] It is important to note that this conclusion does not imply that differences in browser caching effectiveness between W95 and W98 are insignificant. It is also worth noting that the use of HTTP features like conditional GETs, as well as the browser's validation and reload behavior, have changed the amount and kinds of traffic seen by a network cache significantly.

hence cacheability. To do so, we need to perform simulation experiments in which the MPLs for W95 and W98 are equalized.

To study the effect of varying levels of inter-session temporal locality, a tool was developed to artificially superimpose and concatenate individual user sessions to achieve a known and consistent MPL. The individual sessions are all normalized to have a zero start time. The synthesizer then randomly permutes the sequence of individual sessions and selects from it until the desired MPL is achieved; as each session completes, a new session is selected and its timestamps adjusted to begin one second after the previous session completes. The synthesizer continues doing this until the synthetic workload contains the desired total number of requests. Sessions are not replayed unless the desired number of requests exceeds the number of requests in the source trace; permutation was chosen over simple random selection because a recurring session would represent a recurring request pattern not present in our actual traces, possibly artificially inflating Hit rate and Byte hit rate.

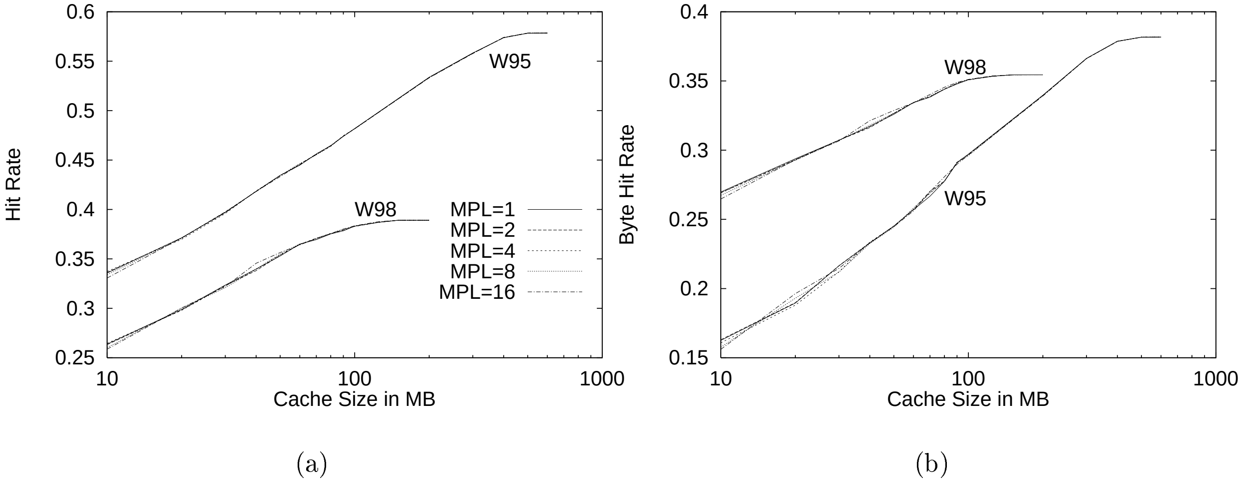


Figure 16: Performance of LRU as a function of cache size under MPL of 1, 2, 4, 8, and 16: (a) H metric (b) B metric.

Since this simulation requires permuting subsets of the original traces, several "batches" of workloads were synthesized, where each batch contained five traces with MPL's of 1, 2, 4, 8, and 16, each trace accessing the same set of documents. Figure 16 shows the document hit rates (H) and byte hit rates (B) for a representative batch, and illustrates that MPL did not significantly effect these metrics.[9] Thus we conclude that our results are not significantly due to differences in MPL between the two traces.

[9] The study in [Bestavros and Cunha 1996] showed that inter-session temporal locality (termed "geographical locality") is key to effective *server* caching—as opposed to the *network* caching discussed in this paper.

5.2.3 Effect of Shared Working Sets

The results we have shown in this section indicate that differences in browser caching alone do not explain the decrease in hit rates in the W98 trace compared to the W95 trace. In addition our results suggest that declines in inter-session locality of reference play a significant role in this decrease in cacheability. This suggests that the working sets of different users have less in common in 1998 than they did in 1995.

This change may be due to: (1) changes in the Web itself (*e.g.,* the number, type, and structure of documents available in 1995 versus 1998), and/or (2) changes in the user population (e.g., more diverse interests, more sophisticated usage, *etc.*) In both cases, the result is the same: the potential for inter-session sharing of working sets has declined from 1995 to 1998.

As evidence of the decline in inter-session sharing (*i.e.,* recurrent requests by different users), we note the following: W95 included 46720 unique document names and our perfect browser cache simulation still required 120236 requests, meaning that on average, a document was requested by 2.57 different users. On the other hand, W98 included 37747 unique document names and our perfect browser cache simulation required only 48111 requests, so in W98 the average document was requested by only 1.27 users—less than half as many.

6 CONCLUSION

In this study we've compared two samples of Web client behavior that are separated in time by three years. Our study is unique because the samples were taken at very different times, but the computing facility and nature of the user population remained the same for both measurements (this also means that our more recent workload measurements should not be assumed to be representative of typical current Web users). Our goal was to discover whether important characteristics of our Web client workloads have changed over the three year time span, and to understand reasons for the changes that have occurred.

The first set of characteristics we examined are the distributional properties of Web files. We studied the set of transferred files because this set influences the properties of network traffic; and we studied the set of unique files because this set can give insight into the set of files that may be present in large network caches.

Both sets continue to show the heavy-tailed property in 1998 that was first identified in 1995. This property is important because it has been suggested as a causal mechanism for the presence of self-similarity in Web traffic [Crovella and Bestavros 1997]. In addition, both sets are well modeled as a hybrid distribution with a lognormal body and a Pareto tail, which was also true in 1995. However, we find that distributionally, the 1998 datasets show a shift toward smaller sizes overall, and lighter tails, than the 1995 datasets. This trend is quite pronounced in the case of the set of unique files, but is only slightly present in the case of the set of transferred files.

We also find that the nature of file popularity, as demonstrated by Zipf-like laws, is different between the two datasets. For the 1995 dataset, the degree of popularity imbalance among transferred files is much greater than for the 1998 dataset. That is, file requests are more evenly spread over the set of unique files in the 1998 dataset than was the case in the 1995 dataset. This result suggests that network caching may be less effective when applied to the 1998 dataset; and in fact we found this to be true in simulation.

Our caching simulations resulted in two main conclusions: first, that cache replacement policies for the 1998 dataset benefit from the use of size less than for the 1995 dataset; and second, that overall effectiveness of network caching is lower in 1998 than in 1995.

Our conclusion regarding cache replacement policies can be understood in terms of file size preferences. When small files are preferred (that is, the covariance of requests per file and file size takes on a large negative value), LFF will tend to outperform LRU under the H metric. This was generally true for the 1995 traces. However, the decrease in the negative covariance of R_i and S_i from 1995 to 1998 means that the advantage of LFF over LRU under the H metric tends to diminish as well. Furthermore, regardless of the nature of the covariance of R_i and S_i, LRU yields better results than LFF under the B metric. Thus, the use of size in cache replacement algorithms used in a network cache needs to be examined carefully, and may be less desirable for our 1998 traces than it was for the 1995 traces.

Our simulations also demonstrate that the 1998 traces show significantly lower hit rates than the 1995 traces. This is a difference that has implications for many current efforts in network caching for the Web. Since this conclusion is important, we explored a number of possible explanations for this effect. First, we showed that the lower hit rates in 1998 were not solely due to improved caching at the clients because even in the presence of perfect caching, hit rates would be much lower for the 1998 traces. Next, we showed that the lower hit rates were not due to lower levels of session interleaving (MPL), because synthetically created variations in session interleaving did not affect hit rates. As a result, we conclude that the inherent potential for caching of requests across sessions is lower in 1998 than it was for 1995.

7 ACKNOWLEDGMENTS

This work was partially supported by NSF research grants CCR-9706685 and CCR-9501822, and by Microsoft Research and Hewlett-Packard Laboratories.

REFERENCES

Abdulla, G., E. A. Fox, and M. Abrams (1997), "Shared User Behavior on the World Wide Web," In *Proceedings of WebNet 97*.

Abrams, M., C. R. Standridge, G. Abdulla, S. Williams, and E. A. Fox (1995), "Caching proxies: limitations and potentials," *The World Wide Web Journal 1*, 1.

Almeida, V., A. Bestavros, M. Crovella, and A. de Oliveira (1996), "Characterizing Reference Locality in the WWW," In *Proceedings of 1996 International Conference on Parallel and Distributed Information Systems (PDIS '96)*, pp. 92–103.

Arlitt, M. F. and C. L. Williamson (1997), "Web Server Workload Characterization: The Search for Invariants," *IEEE/ACM Transactions on Networking 5*, 5, 631–645.

Barford, P. and M. E. Crovella (1998), "Generating Representative Web Workloads for Network and Server Performance Evaluation," In *Proceedings of Performance '98/SIGMETRICS '98*, pp. 151–160.

Bestavros, A., R. L. Carter, M. E. Crovella, C. R. Cunha, A. Heddaya, and S. A. Mirdad (1995), "Application-Level Document Caching in the Internet," In *Proceedings of the Second International Workshop on Services in Distributed and Networked Environments (SDNE'95).*

Bestavros, A. and C. Cunha (1996), "Server-initiated Document Dissemination for the WWW," *IEEE Data Engineering Bulletin 19*, 3–11.

Bolot, J.-C. and P. Hoschka (1996), "Performance Engineering of the World Wide Web: Application to Dimensioning and Cache Design," In *Proceedings of the Fifth Interntional Conference on the WWW*, Paris, France.

Bolot, J.-C., S. Lamblot, and A. Simonian (1997), "Design of Efficient Caching Schemes for the World Wide Web," In *Teletraffic Contributions for the Information Age, Proceedings of the 15th International Teletraffic Congress (ITC-15)*, V. Ramaswami and P. Wirth, Eds., pp. 403–412.

Catledge, L. D. and J. E. Pitkow (1995), "Characterizing browsing strategies in the World-Wide Web," *Computer Networks and ISDN Systems 26*, 6, 1065–1073.

Crovella, M. E. and A. Bestavros (1997), "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes," *IEEE/ACM Transactions on Networking 5*, 6, 835–846.

Crovella, M. E. and L. Lipsky (1997), "Long-Lasting Transient Conditions in Simulations with Heavy-Tailed Workloads," In *Proceedings of the 1997 Winter Simulation Conference*, pp. 1005–1012.

Crovella, M. E., M. S. Taqqu, and A. Bestavros (1998), "Heavy-Tailed Probability Distributions in the World Wide Web," In *A Practical Guide To Heavy Tails*, chapter 1, Chapman & Hall, New York, pp. 3–26.

Cunha, C. A., A. Bestavros, and M. E. Crovella (1995), "Characteristics of WWW Client-based Traces," Technical Report TR-95-010, Boston University Department of Computer Science.

D'Agostino, R. B. and M. A. Stephens, Eds. (1986), *Goodness-of-Fit Techniques*, Marcel Dekker, Inc.

Deng, S. (1996), "Empirical Model of WWW Document Arivals at Access Links," In *Proceedings of the 1996 IEEE International Conference on Communication.*

Feldmann, A. and W. Whitt (1997), "Fitting Mixtures of Exponentials to Long-Tail Distributions To Analyze Network Performance Models," In *Proceedings of IEEE INFOCOM'97*, pp. 1098–1116.

Glassman, S. (1994), "A Caching Relay for the World Wide Web," In *Proceedings of the First International World Wide Web Conference*, pp. 69–76.

Huberman, B. A., P. L. T. Pirolli, J. E. Pitkow, and R. M. Lukose (1998), "Strong Regularities in World Wide Web Surfing," *Science 280*, 5360, 95–97.

Iyengar, A. K., E. A. MacNair, M. S. Squillante, and L. Zhang (1998), "A General Methodology for Characterizing Access Patterns and Analyzing Web Server Performance," In *Proceedings of MASCOTS '98*.

Maltzahn, C., K. J. Richardson, and D. Grunwald (1997), "Performance Issues of Enterprise Level Web Proxies," In *Proceedings of the 1997 ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems*, pp. 13–23.

Mandelbrot, B. B. (1983), *The Fractal Geometry of Nature*, W. H. Freedman and Co., New York.

Manley, S. and M. Seltzer (1997), "Web facts and fantasy," In *Proceedings of the 1997 USENIX Symposium on Internet Technologies and Systems*.

Markatos, E. (1996), "Main Memory Caching of Web Documents," In *Proceedings of the Fifth Interntional Conference on the WWW*.

Muntz, D. and P. Honeyman (1992), "Multi-level Caching in Distributed File Systems or Your cache ain't nuthing but trash," In *Proceedings of the Winter 1992 USENIX*, pp. 305–313.

Murta, C. D., V. Almeida, and W. M. Jr. (1998), "Analyzing Performance of Partitioned Caches for the World Wide Web," In *Proceedings of the Third International WWW Caching Workshop*.

Nishikawa, N., T. Hosokawa, Y. Mori, K. Yoshida, and H. Tsuji (1998), "Memory-Based architecture for distributed WWW caching proxy," *Computer Networks and ISDN Systems 30*, 205–214.

Pederson, S. and M. Johnson (1990), "Estimating Model Discrepancy," *Technometrics* .

Pitkow, J. E. (1997), "Summary of WWW Characterizations," In *Proceedings of the Seventh World Wide Web Conference (WWW7)*.

Williams, S., M. Abrams, C. R. Standridge, G. Abdulla, and E. A. Fox (1996), "Removal Policies in Network Caches for World-Wide Web Documents," In *Proceedings of ACM SIGCOMM '96*.

Zipf, G. K. (1949), *Human Behavior and the Principle of Least-Effort*, Addison-Wesley, Cambridge, MA.