Web Caching and Zipf-like Distributions: Evidence and Implications

Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker.

Abstract—

This paper addresses two unresolved issues about web caching. The first issue is whether web requests from a fixed user community are distributed according to Zipf's law [22]. Several early studies have supported this claim [9], [5], [1] while other recent studies have suggested otherwise [16], [2]. The second issue relates to a number of recent studies on the characteristics of web proxy traces, which have shown that the hit-ratios and temporal locality of the traces exhibit certain asymptotic properties that are uniform across the different sets of the traces [4], [19], [7], [10], [15]. In particular, the question is whether these properties are inherent to web accesses or whether they are simply an artifact of the traces. An answer to these unresolved issues will facilitate both web cache resource planning and cache hierarchy design.

We show that the answers to the two questions are related. We first investigate the page request distribution seen by web proxy caches using traces from a variety of sources. We find that the distribution does not follow Zipf's law precisely, but instead follows a Zipf-like distribution with the exponent varying from trace to trace. Furthermore, we find that there is only (i) a weak correlation between the access frequency of a web page and its size and (ii) a weak correlation between access frequency and its rate of change. We then consider a simple model where the web accesses are independent and the reference probability of the documents follows a Zipf-like distribution. We find that the model yields asymptotic behaviors that are consistent with the experimental observations, suggesting that the various observed properties of hit-ratios and temporal locality are indeed inherent to web accesses observed by proxies.

Finally, we revisit web cache replacement algorithms and show that the algorithm that is suggested by this simple model performs best on real trace data. The results indicate that while page requests do indeed reveal short-term correlations and other structures, a simple model for an independent request stream following a Zipf-like distribution is sufficient to capture certain asymptotic properties observed at web proxies.

Keywords— **caching, World Wide Web, Zipf distribution.**

I. Introduction

DUE to the explosive growth of the web, web proxy caching has recently received considerable attention. It is considered one of the most important techniques for reducing web traffic, which accounts for a large percentage of Internet traffic today. Several researchers have observed that the relative frequency with which web pages are requested follows Zipf's law [22]. Zipf's law states that the relative probability of a request for the i'th most popular page is proportional to $1/i$. Glassman [9] was perhaps the first to use Zipf's law to model the distribution of web page requests, and several other authors have also applied Zipf's law to the distribution of web requests [5], [1]. However, several recent studies have investigated whether the requests do indeed follow Zipf's law and concluded otherwise [16], [2].

One of our goals in this paper is to explore the applicability of Zipf's law to web requests. Using six traces from proxies at academic institutions, corporations and ISPs, we find that *the distribution of page requests generally follows a Zipf-like distribution where the relative probability of a request for the i'th most popular page is proportional to $1/i^\alpha$, with α typically taking on some value less than unity.* The observed value of the exponent α varies from trace to trace. That is, the request distribution does not follow the strict Zipf's law (for which $\alpha = 1$), but instead follows a more general Zipf-like distribution with varying α. Moreover, we find that there is little correlation between the access frequency of a document and its size, and the correlation between the access frequency of a document and its rate of modification varies from very low to none, depending on the traces. These results raise the possibility that, for some purposes, one might be able to sufficiently model web accesses by a simple model that assumes independent references following a Zipf-like distribution and no correlation between request frequency and response size or rate of change.

In looking at web proxy traces, researchers have also investigated how the hit-ratio depends, asymptotically, on the cache size and the number of requests, and have examined the temporal locality of these request streams [4], [19], [7], [10], [15]. Although various studies have used different sets of traces, the following three qualitative asymptotic properties have been identified:

- *For an infinite sized cache, the hit-ratio for a web proxy grows in a log-like fashion*[1] *as a function of the client population of the proxy and of the number of requests seen by the proxy.* Cao et al. observed this property in Digital Equipment Corporation's proxy traces [4], [14], Gribble et al. observed this property in University of California at Berkeley's proxy traces [10], [11] and Duska et al. observed this property in a number of traces from university proxies and ISP proxies [7].
- *The hit-ratio of a web cache grows in a log-like fashion as a function of the cache size.* Many web caching studies reach this conclusion [1], [9], [4], [21], [10], [19], [5], [7].
- *The probability that a document will be referenced k requests after it was last referenced is roughly proportional to $1/k$.* That is, web traces exhibit excellent temporal locality. Of the two studies that investigated temporal locality, Rizzo et al. observed this property in the web proxy traces

Lee Breslau and Scott Shenker are with the Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304. Email: {breslau|shenker}@parc.xerox.com. Pei Cao and Li Fan are with the Computer Science Department, University of Wisconsin, Madison, WI 53706. Email: {cao|lfan}@cs.wisc.edu. Graham Phillips is with the Computer Science Department, University of Southern California, Los Angeles, CA 90089. Email: graham@catarina.usc.edu.

[1] That is, the hit-ratio grows as either the log, or as a small power, of the argument. Given the coarseness of the data, it is impossible to distinguish between a logarithmic and a small power growth.

collected at University of Pisa, Italy [19], while Cao et al. observed this property in the Digital Equipment Corporation's proxy traces [4], [14].

These observations were offered without an underlying explanation. It is not clear whether these properties follow certain inherent characteristics of web accesses or are simply an artifact of the collection of traces studied. The properties are useful for designing caching algorithms and configuring proxy caches, and therefore it is important to understand how general they are.

In the second portion of this paper, we show that the asymptotic behavior of the cache hit-rate and temporal locality are related to the Zipf-like nature of the request stream. More specifically, we show that *if one assumes that web page requests are independent and the probability that a page is accessed follows a Zipf-like distribution then the three asymptotic properties listed above all follow.* Although the assumption that the requests are independent is obviously an over-simplification, our results show that the model is powerful enough to explain the three asymptotic properties as observed in real proxy traces.

The rest of the paper is organized as follows. In Section II we describe the trace data and the resulting frequency distribution of requests. Section III discusses the implications of the Zipf-like behavior, with Section III-A describing the simple model for page requests and Sections III-B, III-C and III-D deriving expressions for cache hit-ratio and page request interarrival-times. We analyze web cache replacement algorithms in Section IV where we compare the LRU and LFU page replacement policies. We conclude in Section V with a short discussion of our results.

II. The applicability of Zipf's Law to web requests

SEVERAL prior studies have investigated the application of Zipf's law to web accesses and arrived at different conclusions.

- In 1994, Glassman et al. [9] traced web accesses from Digital Equipment Corporation's Palo Alto facilities and gathered about 100,000 HTTP requests and found that the request distribution fit Zipf's law, Ω/i, quite well (where Ω is a normalizing constant).
- In 1995, Cunha et al. [5] gathered 500,000 web accesses from the Computer Science Department at Boston University and observed that the requests follow an Ω/i^α distribution where $\alpha = 0.986$, which is quite close to the true Zipf's law.
- In 1996, Almeida et al. [1] showed that web accesses seen by a web server follow Zipf's law. However, the web accesses seen at a web server and those seen at a web proxy are different, because the former includes requests from all users on the Internet while the latter includes only those users from a fixed group.
- In 1998, Nishikawa et al. [16] from Hitachi, Ltd, Japan studied an access log of 2,000,000 requests and found that the request distribution followed the Zipf-like distribution Ω/i^α quite well, but with $\alpha = 0.75$, which is rather far from true Zipf behavior.
- In 1998, Almeida et al. [2] examined requests at a major proxy cache in Brazil and concluded that the request stream did not fit Zipf's law.

There might be several reasons for the conflicting results, including the fact that some of the traces are old and some are quite short. Adding further to the confusion is the phrase "Zipf's law," which in some papers refers to the Ω/i distribution, and in others refers to the Ω/i^α distribution with $\alpha \leq 1$.

To address the issue, we study six traces from academic, corporate and ISP environments, and examine whether the request distribution follows Ω/i or Ω/i^α with $0 < \alpha < 1$. We call Ω/i^α with $0 < \alpha < 1$ a *Zipf-like behavior*, and reserve the phrase Zipf's law for the true Ω/i distribution.

The six traces are:

- **DEC** traces [14]: Digital Equipment Corporation web proxy traces, servicing about 17,000 workstations. We use a 7 day portion of the trace starting from August 29, 1996 and comprising 3,543,968 requests.
- **UCB** traces [11]: HTTP requests gathered from the Home IP service offered by UC Berkeley to its academic community. We use an 18 day portion of the trace starting from November 14, 1996 and comprising 1,907,762 requests.
- **UPisa** traces [18]: HTTP requests made by the users in the Computer Science Department in Universita di Pisa, Italy. We use a portion of a three month trace that includes only HTTP GET requests and URLs that do not include query strings. The trace includes 2,833,624 requests.
- **Questnet** traces [20]: a 7 day trace of HTTP requests seen by the parent proxies at Questnet, a regional ISP in Australia, starting from January 15, 1998. We extract all successful GET requests as seen by the parent proxies. The trace includes 2,885,285 requests.
- **NLANR** traces [17]: a one day trace of HTTP requests to four proxy caches at the National Lab for Applied Networking Research, USA on December 22, 1997. We extract successful GET requests, totaling 1,766,409 requests.
- **FuNet** traces [13]: a 10 day (around mid June, 1998) trace of HTTP requests to proxies at FuNet, a regional network serving the academic communities in Finland. The trace contains 4,815,551 requests.

Our main observations are as follows:

- The distribution of web requests from a fixed group of users follows a Zipf-like distribution, Ω/i^α, very well. The value of α varies from trace to trace, ranging from 0.64 to 0.83.
- The "10/90" rule (i.e. 90% of accesses go to 10% of items), evident in program execution, does not apply to web accesses seen by a proxy. The concentration of web accesses to "hot" documents depends on α, and it takes 25% to 40% of documents to draw 70% of web accesses.
- There is low statistical correlation between the frequency that a document is accessed and its size, though the average size of cold documents (for example, those accessed less than 10 times) is larger than that of hot documents (for example, those accessed more than 10 times).
- The statistical correlation between a document's access

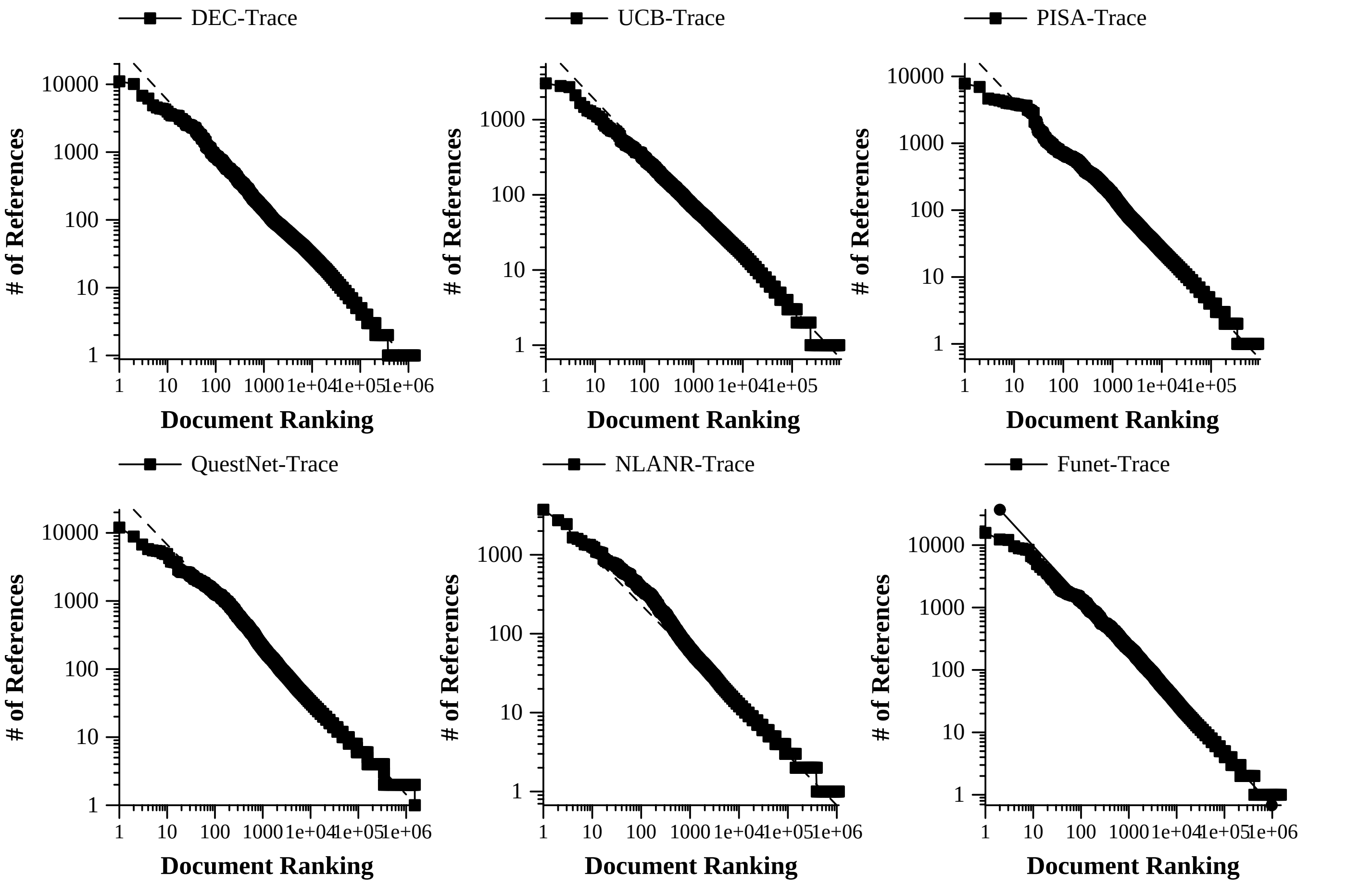

Fig. 1. Frequency of document accesses versus document ranking.

TABLE I

VALUES OF α FOR EACH TRACE.

DEC	UPisa	FuNet	UCB	Questnet	NLANR
0.77	0.78	0.83	0.69	0.73	0.64

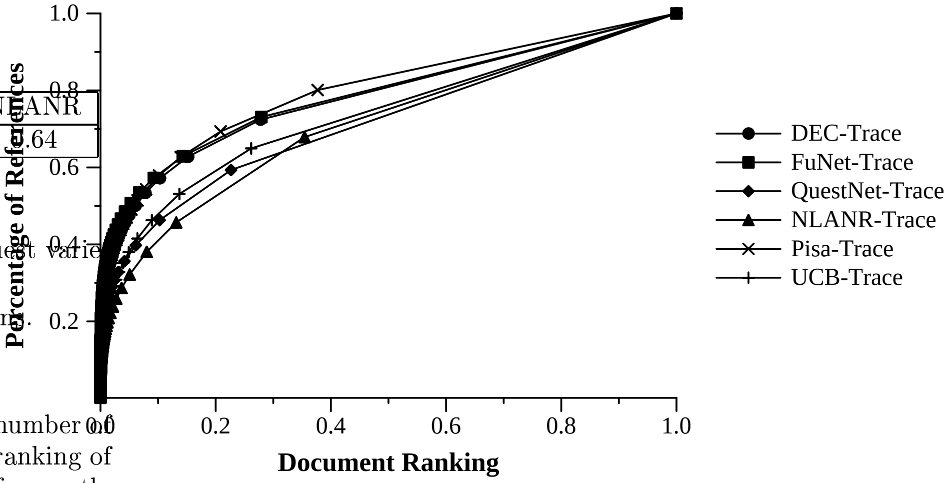

Fig. 2. Cumulative distribution of requests to documents.

frequency and its average modifications per request varies from trace to trace, but is generally quite low.

The results are detailed in the following sections.

A. Is the distribution of web requests Zipf-like?

Figure 1 shows, for each of the six traces, the number of times that a URL has been accessed versus the ranking of the URL in the trace, where rank 1 is the most frequently accessed URL. Note that both axes in the figure are in log scale.

From the plots, one can observe that most of the curve fits a straight line reasonably well. The straight line on the log-log scale implies that the request frequency is proportional to $1/i^{\alpha}$. The values of α for the six traces are shown in Table I; they are obtained using MatLab's curve-fitting tool, excluding the top 100 documents. For traces from a homogeneous environment, as in a corporation (DEC) or academic department (Pisa), α appears to be larger, centering around 0.8. For traces from a more diversified user population (UCB) or those that are filtered by first-level proxies (QuestNet and NLANR), α appears to be smaller, centering around 0.7.

Perhaps the biggest impact of α lies in the concentration of web requests to hot documents. For Zipf-like distributions, the cumulative probability that one of the top k page is accessed is given asymptotically by $\phi(k) = \sum_{i=1}^{k} \cdot \Omega/i^{\alpha} \approx \Omega k^{1-\alpha}/(1-\alpha)$. Because $\Omega \approx (1-\alpha)/N^{1-\alpha}$, where N is

TABLE II
CORRELATION COEFFICIENT BETWEEN THE FREQUENCY OF ACCESSES TO A DOCUMENT AND ITS SIZE.

DEC	UPisa	FuNet	UCB	Questnet	NLANR
0.04	-0.08	0.003	-0.04	-0.02	-0.09

TABLE III
CORRELATION COEFFICIENT BETWEEN DOCUMENT CHANGE RATE AND ACCESS FREQUENCY.

DEC	UPisa	FuNet	UCB	Questnet	NLANR
-0.19	-0.27	0.005	0.002	-0.03	-0.08

the total number of web documents, $\phi(k) \approx (k/N)^{1-\alpha}$. Because $k/N < 1$ for all meaningful k, a larger α increases $\phi(k)$, meaning more requests are concentrated on a few hot documents.

We examine the the cumulative distribution of requests to popular documents in each trace. Figure 2 shows the cumulative probability of access for the top r% of documents in each trace. The figure shows that the top 1% of the documents account for about 20% to 35% of all requests seen by the proxy, and the top 10% of the documents account for about 45% to 55% of all requests. However, it takes 25% to 40% of all documents to account for 70% of the requests, and it takes 70% to 80% of documents to account for 90% of requests. In other words, the "10/90" rule, evident in program execution, unfortunately does not apply to web accesses. This is why, in practice, web caches rarely achieve hit ratios of 90%, and no satellite-based broadcast scheme can reduce Internet traffic by an order of magnitude.

B. Correlation between access frequency and document size

A natural question is whether there is any correlation between the frequency of access to a document and its size. The answer affects the design of caching algorithms and web proxy benchmarks. Figure 3 shows the average size of the documents that are accessed exactly n times in the trace, with the x-axis showing n. Again, both axes are in log scale. Due to space constraints we show results of three traces; the rest are similar and can be found in [3]. The figure shows that there are many popular documents that are larger than 15KB, which is the average size of documents that are accessed only once. Furthermore, across n, there does not seem to be strong correlation between document size and n, though the average size of popular documents is smaller than that of unpopular documents.

We also calculated the correlation coefficient between the access frequency and document size for the traces, shown in Table II. The numbers show that the correlation, if any, is weak and can be ignored.

C. Correlation between access frequency and change rate

Another question of interest is whether popular documents change less often than unpopular ones. The answer affects designs of web cache consistency mechanisms [12].

From the traces, we observe the changes in a document by detecting changes in the last-modified-time of each response or changes in the document size of each response. Only two of our traces, DEC and UCB, come with the last-modified-time information. For each document, we then measure the "change rate" r as the ratio between the number of observed changes and the number of accesses. The observed change rate under-estimates the actual change rate, because multiple changes between requests are not picked up by the trace. Because of this potential skew, we do not measure such things as the number of changes per day for web pages since these numbers would be misleading.

Figure 4 plots the average change rate of the documents as a function of the number of times they are accessed. Only the x-axis is in log scale. We show results of three traces here; the rest are similar and can be found in [3]. The graphs show that while many popular documents change infrequently (for example, less than 10 user requests per change), some popular documents change very frequently. Thus, there is no strong evidence that popular documents change infrequently. We again calculate the correlation coefficient between the document change rate and its access frequency. The results are shown in Table III. Though the correlation coefficients vary depending on the traces, they are small in general. Therefore, in the design of web cache consistency schemes, it is best to assume that there is no correlation between document popularity and its rate of change.

D. Request distribution to web servers

We have also investigated the distribution of requests to web servers. Figure 5 shows the number of times that a server has been accessed versus the ranking of the server in the trace, where rank 1 is the most accessed server. Both axes are in log scale. We show only three traces here, with the rest in [3]. In the case of the UCB trace, the Zipf-like distribution does not characterize server accesses. We also found that no single web server contributes to most of popular pages, and popular web pages are spread almost evenly across hot web servers. For more details see [3].

III. IMPLICATIONS OF ZIPF-LIKE BEHAVIOR

WE now investigate the implications of the Zipf-like behavior observed in the traces. First, we define a simple model for web requests where the requests are independent and distributed according to a Zipf-like behavior. We then derive expressions for the hit-ratios and request interarrival-times.

A. The model

Consider a cache that receives a stream of requests for web pages. Let N be the total number of web pages in the universe. Let $P_N(i)$ be the conditional probability that, given the arrival of a page request, the arriving request is made for page i. Let all the pages be ranked in order of

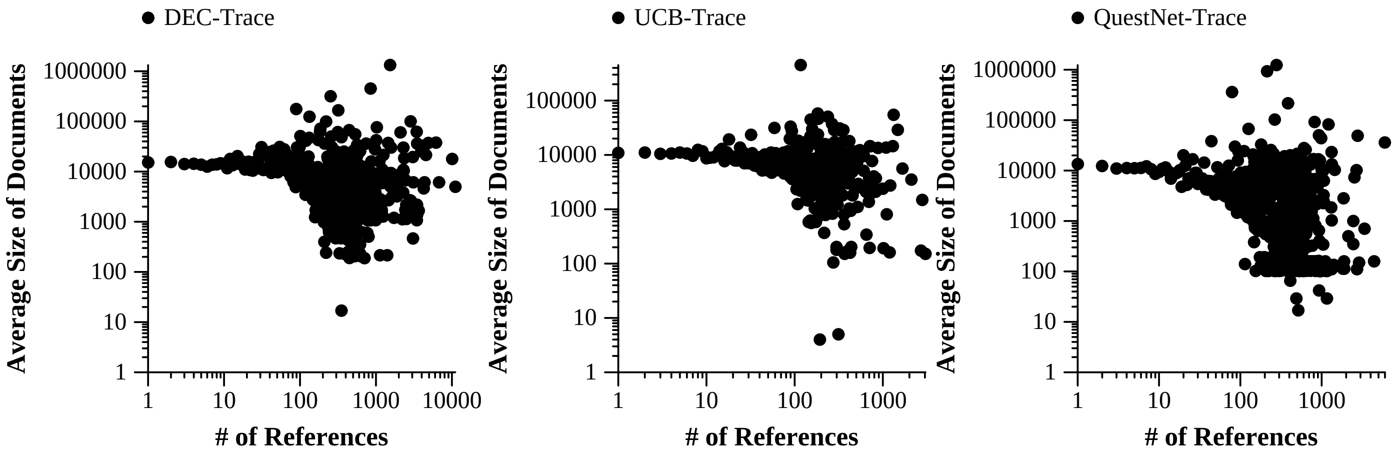

Fig. 3. Average size of a document versus access frequency.

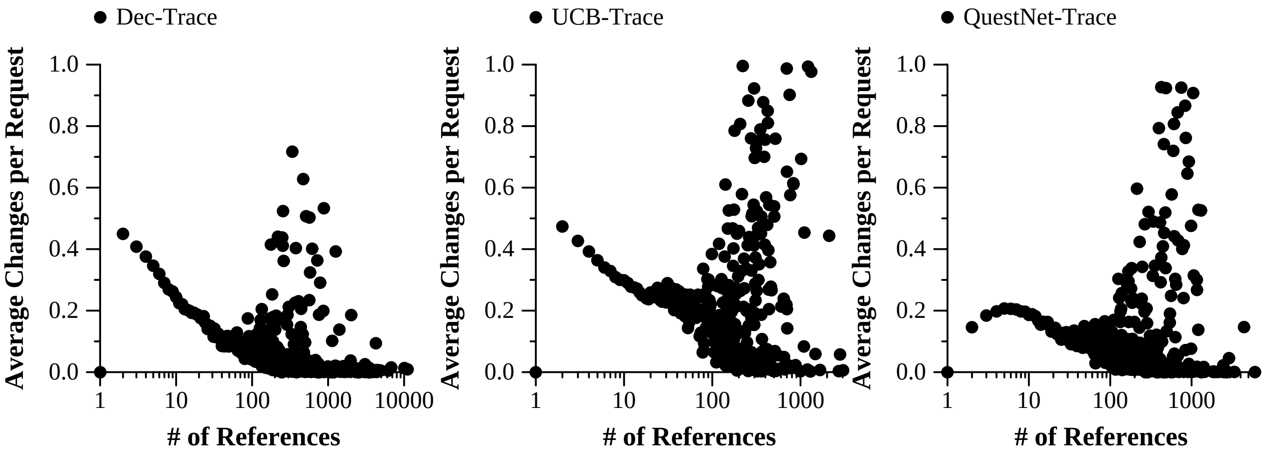

Fig. 4. Change rate of a document versus access frequency.

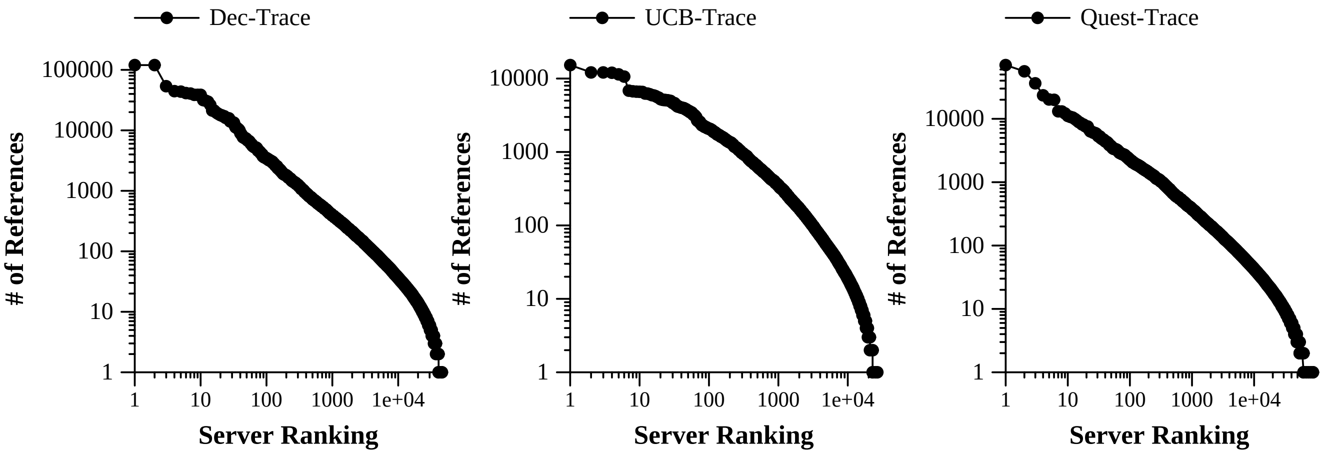

Fig. 5. Frequency of server accesses versus server ranking.

their popularity where page i is the i'th most popular page. We assume that $P_N(i)$, defined for $i = 1, 2, \ldots, N$, has a "cut-off" Zipf-like distribution given by

$$P_N(i) = \frac{\Omega}{i^\alpha},$$

where

$$\Omega = \left(\sum_{i=1}^{N} \frac{1}{i^\alpha} \right)^{-1}$$

The true Zipf's law [22] has $\alpha = 1$ but in this paper we consider a broader class of distribution functions with exponents in the range $0 < \alpha \leq 1$. Each page request is drawn independently from the Zipf distribution, so we are neglecting any other source of correlations in the request stream. We additionally assume that, over the course of time, no pages are invalidated by the cache.

We acknowledge that this model is unrealistic because it assumes that requests are independent. However, the model is tractable, and our question here is whether the model is sufficient to derive an understanding of the asymptotic properties of hit-ratios and request interarrival-times.

In the following three subsections, we calculate the hit-ratio for the cache in two limiting cases and also the interarrival-times in a limiting case.

B. Infinite cache, finite request stream

We first consider the case where the cache has unlimited storage, so that all previously requested pages remain in the cache. In this case, we consider a finite request stream of R requests, and wish to determine the probability that the next request, the $R+1$'st request, is a request for a page that already resides in the cache. The hit-ratio $H(R)$ can be calculated as follows. If the $R+1$'st request is for page i then the probability that this page is in the cache is given by $(1-(1-P_N(i))^R)$. Thus, we have:

$$H(R) = \sum_{i=1}^{N} P_N(i) \left(1 - (1 - P_N(i))^R\right) \qquad (1)$$

For $\alpha = 1$ the asymptotic behavior of the hit-ratio is $H(R) \approx \ln R$, while for other values of α the asymptotic behavior of the hit-ratio is $H(R) \approx R^{(\frac{1}{\alpha}-1)}$. This asymptotic behavior can be seen by approximating the function $f(i) = (1-(1-\frac{\Omega}{i})^R)$, for $1 \ll R \ll N$, by:

$$f(i) = \begin{cases} 1, & \text{if } i^\alpha \leq R\Omega \\ 0, & \text{otherwise} \end{cases}$$

Then we have

$$H(R) = \sum_{i=1}^{N} \frac{\Omega}{i^\alpha} f(i) \approx \sum_{i=1}^{(R\Omega)^{\frac{1}{\alpha}}} \frac{\Omega}{i^\alpha}$$

For $\alpha = 1$, $H(R) \approx \Omega \ln(R\Omega)$, while for $0 < \alpha < 1$, $H(R) \approx \frac{\Omega}{1-\alpha}(R\Omega)^{(\frac{1}{\alpha}-1)}$. Note that when $R > N$ this approximation is no longer valid because $H(R)$ must approach unity while these expressions diverge for large R. This result that the hit-ratio grows logarithmically or like a small power is qualitatively consistent with previously observed behavior [10], [4], [7].

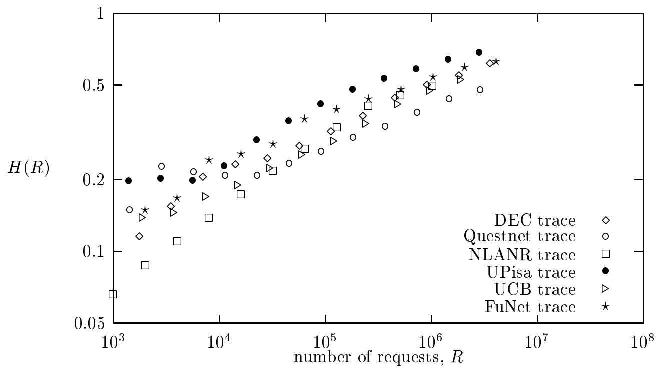

Fig. 6. Hit-ratio $H(R)$ as a function of the number of requests.

TABLE IV

VALUES OF α FROM LINEAR FIT OF $\log H(R)$ VERSUS $\log R$.

DEC	UPisa	FuNet	UCB	Questnet	NLANR
0.83	0.84	0.84	0.83	0.88	0.90

Figure 6 shows the hit-ratio $H(R)$ as observed in the traces as a function of R. The data for each trace on this log-log plot is roughly linear, thereby suggesting that $H(R) \approx R^\beta$ for some β (where the power varies from trace to trace). We fitted each of these log-log data sets to a straight line to obtain an estimate for β. We then used the fact that the model predicts that $H(R) \approx R^{(\frac{1}{\alpha}-1)}$ to compute $\alpha = \frac{1}{1+\beta}$. Table IV displays these α values. The agreement with the α values computed directly from the page request distribution (in Table I) is good, although the Questnet and NLANR results are not as accurate as the others.

C. Finite cache, infinite request stream

In this section we consider a finite cache with a capacity of C web pages subject to an infinitely long request stream. We assume that the cache can hold C web pages regardless of the size of each web page. Furthermore, we assume that the cache holds the C most popular pages as indicated by the ordering i. This ordering is determined by measuring the request frequency for each page, which is equivalent to assuming that the cache has a Perfect-LFU page removal policy (see Section IV).

We are interested in the asymptotic hit-ratio $H(C)$, which is given by:

$$H(C) = \sum_{i=1}^{C} P_N(i)$$

In the case $\alpha = 1$, then asymptotically $H(C) \approx \ln C$, and for $0 < \alpha < 1$, then asymptotically $H(C) \approx C^{1-\alpha}$. This result is qualitatively consistent with previously observed behavior that the hit-ratio increases logarithmically or as a

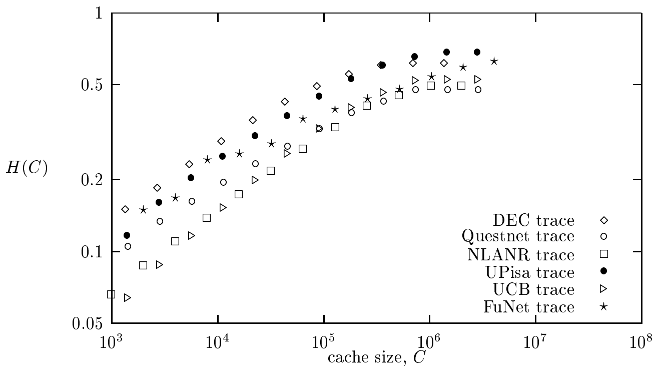

Fig. 7. Hit-ratio $H(C)$ as a function of cache size.

TABLE V

VALUES OF α FROM LINEAR FIT OF $\log H(C)$ VERSUS $\log C$.

DEC	UPisa	FuNet	UCB	Questnet	NLANR
0.75	0.72	0.78	0.66	0.76	0.69

small power as a function of cache size [1], [9], [4], [21], [10], [19], [5], [7]. These references do not claim any particular form for the asymptotic behavior, but their data appears to grow in a log-like fashion in the asymptotic regime.

Figure 7 displays the hit-ratio for the various web cache traces on a log-log plot. Note that some of the graphs for the traces flatten out for large R. This behavior is expected because, for finite sized traces, the hit-ratio will reach a maximum and then remain constant when the cache size becomes larger than the trace file. Thus, this flattening out behavior is a limitation of the size of the trace file rather than an indication of the true asymptotic behavior.

The hit-ratio shown on the log-log scale is roughly linear. As with the $H(R)$ data, this suggests that $H(C) \approx C^{\beta}$ for some β (where the power varies from trace to trace). Using the prediction that $H(C) \approx C^{1-\alpha}$, we computed a linear fit of each log-log data set to find β and then computed $\alpha = 1 - \beta$. When fitting the straight line we excluded all data with $R > 10^6$ to reduce the limitation of the trace file size. The α values are shown in Table V. The agreement with the α values computed directly from the page request distribution (in Table I) is remarkably good. Thus, the simple analytical model has been successful in predicting the asymptotic behavior of $H(R)$ and $H(C)$.

D. Page request interarrival-times

We now investigate the distribution of times between requests for a given page.

Let us consider an infinite arrival stream and consider a request for a given page i. The quantity of interest is the probability distribution $d(k)$ that the next request for that page is k requests later (*i.e.*, that the request for page i is followed by $k-1$ requests for pages other than page i, followed by another request for page i). Assuming that

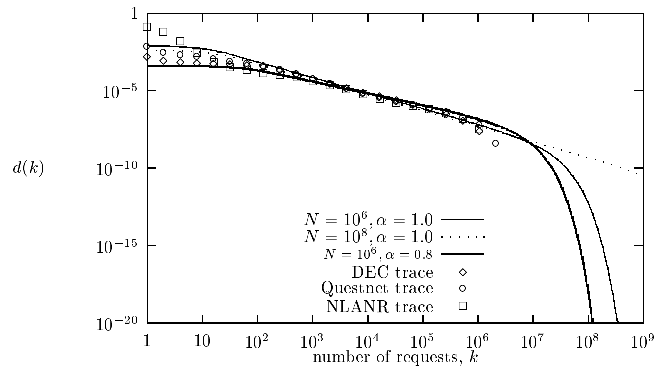

Fig. 8. Distribution of request interarrival-times, $d(k)$.

page requests are independent, we find that

$$d(k) = \sum_{i=1}^{N} (P_N(i))^2 \left(1 - P_N(i)\right)^{k-1} \tag{2}$$

For $\alpha = 1$ we have $\Omega \approx \frac{1}{\ln N}$ and therefore[2]

$$d(k) \approx \frac{1}{k \ln N}\left(\left(1 - \frac{1}{N \ln N}\right)^k - \left(1 - \frac{1}{\ln N}\right)^k\right)$$

Note that for $N \ln N \gg k \gg \ln N$, $d(k) \approx 1/(k \ln N)$.

For $0 < \alpha < 1$ the asymptotic behavior of this expression is less tractable, but as shown in Fig. 8 the behavior in the intermediate regime $N \ln N \gg k \gg \ln N$ is qualitatively similar to the $\alpha = 1$ case, but with different slopes in the linear regime. The model's predictions, in the intermediate regime, are qualitatively consistent with data observed at operational web servers [19], [4].

We have also examined the request interarrival behavior in our traces. Figure 8 shows a plot of the probability distribution for page request interarrival-times $d(k)$ produced by our model and the distribution for three cache traces (we did not show the remaining three traces to keep the graph readable). The agreement between the simple model and the trace data is quite good in the intermediate regime.

IV. Cache replacement algorithms

We next consider whether the simple model introduced in the previous section can be used to improve cache replacement strategies. The model that we have discussed so far is called an *independent reference model* [8] in the early operating system paging studies [6]. It is well known in the operating system caching community that if (i) the requests are independent and have a fixed probability and (ii) the pages have the same size, then the optimal replacement algorithm is to keep those pages with the highest probabilities in the cache [8]. In practice, an online algorithm detects those documents by keeping track of the number of references to each document and keeping

[2] The approximation for $d(k)$ is made by replacing the sum with an integral in equation (2), substituting $P_N(i) = \Omega/i$, then substituting $u = 1 - \Omega/i$ and finally noting that $du = \Omega/i^2 di$.

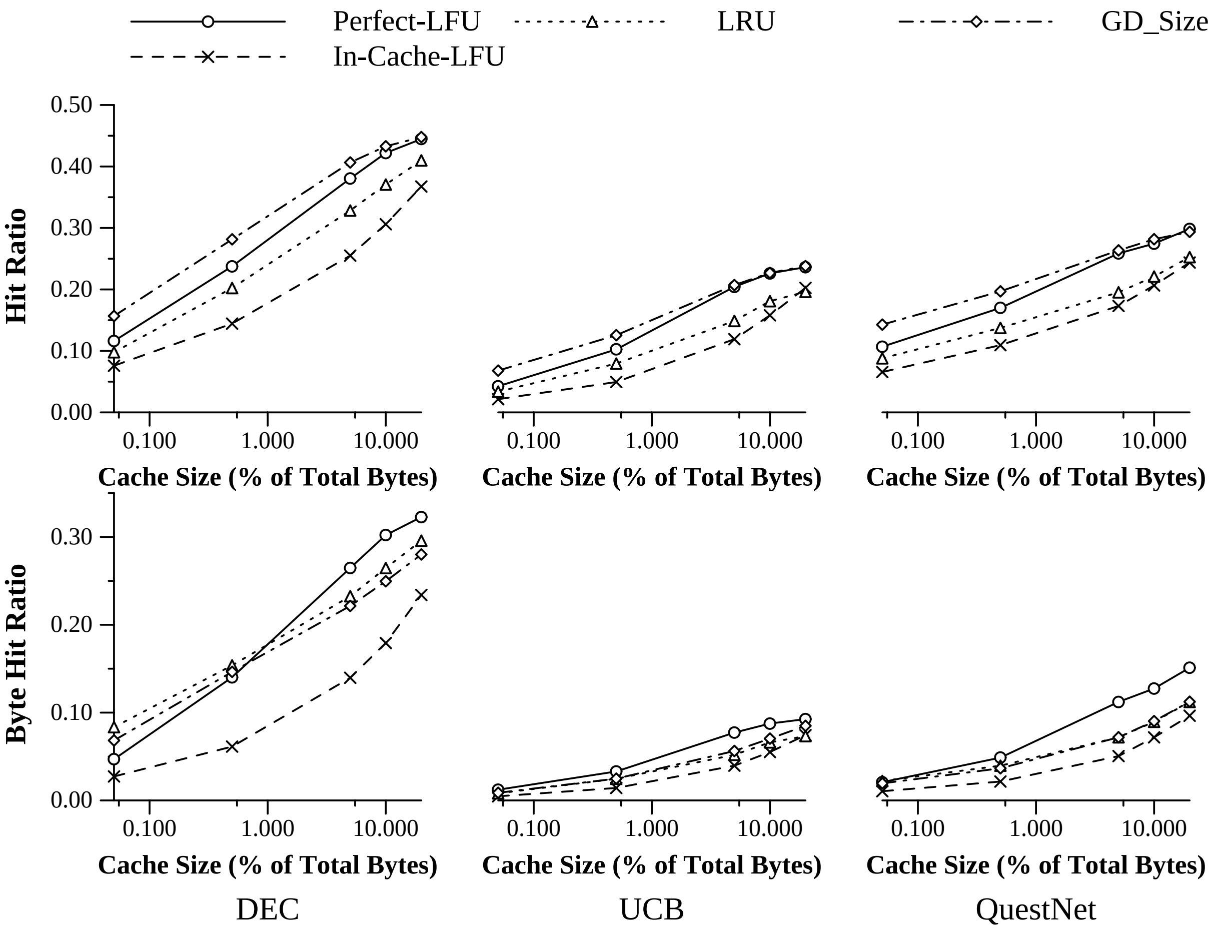

Fig. 9. Hit-ratio and byte hit-ratio for four algorithms for a DEC, UCB and QuestNet traces.

those documents with the highest reference count in the cache. In other words, the best online algorithm under the independent reference model is the Least-Frequently-Used (LFU) algorithm.

However, there are two versions of LFU that are often confused in the literature: In-Cache-LFU, and Perfect-LFU. To make a clear distinction between the two policies, Perfect-LFU remembers page counters even when a page is evicted from the cache, while In-Cache-LFU removes the page counter together with the evicted page. Clearly, Perfect-LFU incurs more overhead and is less practical than In-Cache-LFU.

The question we wish to answer is: Which of the four replacement algorithms—Perfect-LFU, In-Cache-LFU, LRU and GD-Size—performs the best in terms of hit-ratio? Note that LRU is the most widely-used web cache replacement algorithm, while GD-Size is a new algorithm that takes both document size and locality into account and was shown to outperform existing algorithms in terms of hit-ratio [4]. We answer the above question using trace-driven simulations.

Figure 9 shows the hit-ratios and byte hit-ratios for the four replacement algorithms for the DEC, UCB and QuestNet traces. Results from other traces are similar to those in the figure.

Figure 9 shows that, as predicted by the independent reference model, Perfect-LFU performs best in terms of byte hit-ratios in most cases. GD-Size still performs best in terms of hit-ratios for small cache sizes because GD-Size favors small documents over large ones. When cache sizes are large, Perfect-LFU performs comparably to GD-Size in hit-ratio and much better in byte hit-ratio. The figure also shows that In-Cache-LFU performs the worst and consequently is a poor choice for web cache replacement algorithms.

The main drawback of Perfect-LFU is that it requires additional space to maintain the counts for documents that are evicted from the cache. In addition, it fails to take document size into account.

Our goal is not the design of replacement policies, but rather to observe that if temporal locality (due to correlations between references) is the dominant effect then LRU performs the best. However, because LRU did not perform as well as Perfect-LFU, it may suggest that the temporal locality effects may not be important when analyzing qualitative aspects of web caching performance.

For quantitative predictions of web caching performance, it may be important to consider correlations in the request stream. In particular, Almeida et al. compared plots of miss-ratio for a synthetic workload having a Zipf distribution against plots of miss-ratio for real workloads and concluded that the Zipf model was inaccurate because it did not capture the locality of reference in the real request stream [1]. In this paper we looked at the asymptotic properties of the hit-ratio and request interarrival-times, and perhaps in the asymptotic regime the correlations focused on in [1] are not so relevant.

V. Discussion

In this paper we first showed evidence that web requests follow a Zipf-like distribution. We then introduced a simple model for web requests, where requests are independent and distributed according to a Zipf-like distribution, and showed that this simple model can explain the asymptotic behavior for three properties that are observed in real web cache traces. Our results suggest that these properties found in many studies are perhaps inherent to web access streams and not an artifact of the traces studied.

Our results also suggest that a simple model for web requests may be sufficient to understand certain asymptotic properties of cache performance. Our model has many limitations. For instance, the model does not consider document modifications, nor does it consider the cache's replacement policy, both of which no doubt play an important role in a cache's performance. Nonetheless, we think the simple model presented here may be worthwhile for studying the asymptotic behavior of various quantities. We are currently trying to improve the model and gather more traces to validate it.

References

[1] Virgilio Almeida, Azer Bestavros, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the WWW. In *IEEE International Conference in Parallel and Distributed Information Systems*, Miami Beach, Florida, USA, December 1996. http://www.cs.bu.edu/groups/oceans/papers/Home.html.

[2] Virgilio Augusto F. Almeida, Marcio Anthony G. Cesirio, Rodrigo Fonseca Canado, Wagner Meira Junior, and Cristina Duarte Murta. Analyzing the behavior of a proxy server in the light of regional and cultural issues. http://www.anades.dcc.ufmg.br/paperSubmetidos/ webcache/cultural/, 1998.

[3] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and zipf-like distributions: Evidence and implications. Technical report, University of Wisconsin-Madison, Department of Computer Science, 1210 West Dayton Street, July 1998. http://www.cs.wisc.edu/~cao/papers/zipf-implications.html.

[4] Pei Cao and Sandy Irani. Cost-aware WWW proxy caching algorithms. In *Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems*, pages 193–206, December 1997. http://www.cs.wisc.edu/~cao/publications.html.

[5] Carlos Cunha, Azer Bestavros, and Mark Crovella. Characteristics of WWW client-based traces. Technical Report TR-95-010, Boston University, Computer Science Dept., Boston, MA 02215, USA, April 1995. http://www.cs.bu.edu/groups/oceans/papers/Home.html.

[6] Peter J. Denning. Working sets past and present. *IEEE Transaction on Software Engineering*, SE-6(1):64–84, January 1980.

[7] Bradley M. Duska, David Marwood, and Michael J. Feeley. The measured access characteristics of world-wide-web client proxy caches. In *Proceedings of USENIX Symposium on Internet Technology and Systems*, December 1997.

[8] Edward G. Coffman, Jr. and Peter J. Denning. *Operating Systems Theory*. Prentice-Hall, Inc., 1973.

[9] Steven Glassman. A caching relay for the world wide web. In *First International Conference on the World-Wide Web*, CERN, Geneva, Switzerland, May 1994. http://www1.cern.ch/WWW94/PrelimProcs.html.

[10] Steven Gribble and Eric Brewer. System design issues for internet middleware services: Deductions from a large client trace. In *Proceedings of the 1997 Usenix Symposium on Internet Technologies and Systems*, Monterey, California, USA, December 1997. http://HTTP.CS.Berkeley.EDU/~gribble/.

[11] Steven Gribble and Eric Brewer. UCB home IP HTTP traces. http://www.cs.berkeley.edu/ gribble/traces/index.html, June 1997.

[12] James Gwertzman and Margo Seltzer. World-wide web cache consistency. In *Proceedings of the 1996 USENIX Technical Conference, San Diego, CA*, January 1996.

[13] Pekka Jarvelainen. Personal communication. mailto:pj@csc.fi, 1998.

[14] T. M. Kroeger, J. Mogul, and C. Maltzahn. Digital's web proxy traces. ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html, August 1996.

[15] Thomas M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. Exploring the bounds of web latency reduction from caching and prefetching. In *Proceedings of USENIX Symposium on Internet Technology and Systems*, December 1997.

[16] Norifumi Nishikawa, Takafumi Hosokawa, Yasuhide Mori, Kenichi Yoshidab, and Hiroshi Tsujia. Memory-based architecture for distributed WWW caching proxy. In *The 7th WWW Conference*, April 1998. http://www7.conf.au/programme/fullpapers/1928/com1928.htm.

[17] National Lab of Applied Network Research. Sanitized access log. ftp://ircache.nlanr.net/Traces/, July 1997.

[18] Luigi Rizzo. Web proxy traces. http://info.iet.unipi.it/ luigi/proxy-traces/, May 1997.

[19] Luigi Rizzo and Lorenzo Vicisano. Replacement policies for a proxy cache. Technical Report RN/98/13, University College London, Department of Computer Science, Gower Street, London WC1E 6BT, UK, 1998. http://www.iet.unipi.it/~luigi/caching.ps.gz.

[20] Julianne Weekers. Personal communication. mailto:julianne@cc.uq.edu.au, January 1998.

[21] Stephen Williams, Marc Abrams, Charles Standridge, Ghaleb Abdulla, and Edward Fox. Removal policies in network caches for WWW documents. In *ACM SIGCOMM '96*, August 1996. http://www.acm.org/sigcomm/sigcomm96/program.html.

[22] George Kingsley Zipf. *Relative frequency as a determinant of phonetic change*. Reprinted from the Harvard Studies in Classical Philiology, Volume XL, 1929.