On The Implications of Zipf's Law for Web Caching

Lee Breslau
Pei Cao
Li Fan
Graham Phillips
Scott Shenker

Technical Report #1371

April 1998

On the Implications of Zipf's Law for Web Caching

Lee Breslau
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
breslau@parc.xerox.com

Pei Cao
University of Wisconsin, Madison
1210 West Dayton Street
Madison, WI 53706 USA
cao@cs.wisc.edu

Li Fan
University of Wisconsin, Madison
1210 West Dayton Street
Madison, WI 53706 USA
lfan@cs.wisc.edu

Graham Phillips
University of Southern California
University Park
Los Angeles, CA 90089 USA
graham@catarina.usc.edu

Scott Shenker
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304 USA
shenker@parc.xerox.com

Abstract

Recently, a number of studies on characteristics of Web proxy traces have shown that the hit-ratios of the traces exhibit certain properties that are uniform across the different sets of the traces [CI97, RV98, DMF97, GB97, KLM97]. An explanation for these phenomena has eluded researchers and it is not clear whether the properties are inherent to Web accesses or particular to the set of traces studied.

In this paper, we show that if one assumes that the references in the Web access stream are independent and the reference probability of the documents follow Zipf's law then the observed properties follow from Zipf's law. We revisit Web cache replacement algorithms and show that the algorithm that is suggested by Zipf's law performs best. Finally, we investigate the drift in the cache's hot set as a function of time.

1 Introduction

Due to the explosive growth of the Web, Web proxy caching has recently received considerable attention. In particular, several studies have analyzed Web proxy traces and investigated the relationship between cache hit-ratio and cache sizes, the relationship between hit-ratio and the number of requests and the temporal locality of request streams [CI97, RV98, DMF97, GB97, KLM97]. Although various studies have used different set of traces, the following three properties have been identified:

- *Under an infinite cache size, the hit-ratio for a Web proxy is proportional to the log of the client population of the proxy and the log of the number of requests seen by the proxy.* Cao et al. observed this property in Digital Equipment Corporation's proxy traces [CI97, DEC96], Gribble et al. observed this property in University of California at Berkeley's proxy traces [GB97, UCB96] and Duska et al. observed this property in a number of traces from university proxies and ISP proxies [DMF97].

- *The hit-ratio of a Web cache is proportional to the log of the cache size.* Many Web caching studies reach this conclusion [ABCdO96, Gla94, CI97, WAS+96, GB97, RV98, CBC95, DMF97].

- *The probability that a document will be referenced k requests after it was last referenced is 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 collected at University of Pisa, Italy [RV98], while Cao et al. observed this property in the Digital Equipment Corporation's proxy traces [CI97, DEC96].

An explanation for these phenomena, however, has eluded researchers. 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, configuring proxy caches, etc., and therefore it is important to understand them.

In recent years there have been many studies on *page request distribution*, that is, the relative frequency with which Web pages are requested. Numerous studies have found that this distribution follows Zipf's law. Zipf's law states that the relative probability of a request for the i'th most popular page is inversely proportional to i. Perhaps the first use of Zipf's law to model the distribution of Web page requests is by Glassman in [Gla94]. Since then, several authors have shown the applicability of Zipf's law for modelling the requests of Web pages [CBC95, ABCdO96, WAS+96]. Cunha et al. found that the request probability for a Web cache trace, when fitted with a curve of the form i^α, yields a curve with an exponent of $\alpha = -0.982$, which is very close to the value of $\alpha = -1$ in Zipf's law.

In this paper, we show that *if one assumes that Web page requests are independent and the probability that a page is accessed follows Zipf's law then the three properties listed above all follow.* Although the assumption that the requests are independent is traditionally considered an over-simplification, our results show that the model is powerful enough to explain the three properties as observed in real proxy traces.

The rest of the paper is organized as follows. Section 2 describes the Zipf model that is used in sections 3, 4 and 5 to develop expressions for cache hit-ratio and page request interarrival times. With the Zipf model in mind, we revisit Web cache replacement algorithms in section 6 where we compare the LRU and LFU page replacement policies. Finally, in section 7 we investigate how the hot set of a Web cache changes from day to day.

2 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 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 distribution given by

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

where

$$\Omega = \left(\sum_{i=0}^{N} \frac{1}{i}\right)^{-1}$$

Each page request is drawn independently from the Zipf distribution, so we are neglecting any other source of correlations in the request stream. Moreover, we assume that, over the course of time, no pages are invalidated by the cache. We realize that this is an artificial model, but our goal here is to ask what results we can obtain from such a simple assumption.

We calculate the hit-ratio for the cache in two extreme limiting cases and the interrival times in a limiting case, which we consider in the following three sections.

3 Infinite Cache, Finite Request Stream

In this section we investigate the cache hit-ratio as a function of the number of requests and show that the observed property that the function is logarithmic is derivable from the Zipf model.

We consider the case where the cache has unlimited storage, so 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 $R+1$'th 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 $R+1$'th 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:

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

The asymptotic behavior of the hit-ratio is $H(R) \approx \Omega \ln R$. This behavior can be seen more directly by approximating the sum for $H(R)$ by assuming the function $f(i) = (1-(1-\frac{\Omega}{i})^R)$ is given, for $1 \ll R \ll N$, by:

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

Then we have

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

We found that the approximation $\Omega \ln(R\Omega)$ underestimates $H(R)$ and that the approximation $\tilde{H}(R) = \Omega \ln R$ is more accurate. Note that when $R \approx N$ this approximation is no longer valid because $H(R)$ approaches unity while $\tilde{H}(R)$ is unbounded. This result that the hit-ratio is logarithmic is consistent with previously observed behavior [GB97, CI97, DMF97].

Figure 1 shows the hit-ratio $H(R)$, the approximate hit-ratio $\tilde{H}(R)$ and the hit-ratio for a *cache trace* as a function of $\ln R$. The *cache trace* data was collected by Pei et al. using a cache simulator and a Web cache trace file [CI97]. Figure 4 of [CI97] actually plots the hit-ratio as a function of group size, but in the simulation the number of page requests is proportional to the group size.

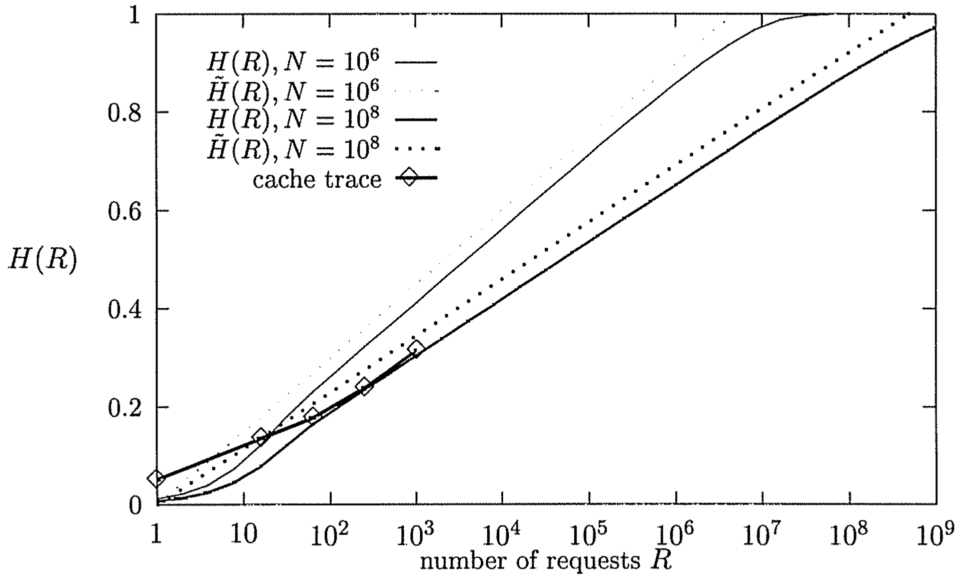

Figure 1: Hit-ratio as a function of the number of requests.

4 Finite Cache, Infinite Request Stream

In this section we investigate the cache hit-ratio as a function of the size of the cache and show that the observed property that this function is logarithmic is also derivable from the Zipf model.

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. The ordering can be determined by measuring the request frequency of each page, which is equivalent to assuming that the cache has a Perfect-LFU page removal policy (see section 6 for a discussion of Perfect-LFU).

We are interested in the asymptotic hit-ratio $H(C)$, which can be calculated to be:

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

This result is consistent with previously observed behavior that the hit-ratio increases logarithmically as a function of cache size [ABCdO96, Gla94, CI97, WAS$^+$96, GB97, RV98, CBC95, DMF97]. For example, Glassman showed this result by using a cache simulator and modelling the requests by a Zipf distribution [Gla94]. Rizzo et al. also published results of hit-ratio as a function of cache size [RV98]. Although they do not compare their graphs to logarithmic functions, we replotted their graphs against a logarithmic scale and found that the graphs were indeed logarithmic.

We also provide empirical results for the hit-ratio observed in several trace files. Figure 2 shows the hit-ratios for Web caches using the LRU replacement algorithm for the client traces gathered at Boston University, traces gathered at Virgnia Tech, and two groups of traces, DEC-U1 and DEC-U2, that are gathered at Digital Equipment Corporation. (For details of the traces see [CI97].) The figure shows that the hit ratio clearly grows proportionally with

$$\ln C$$

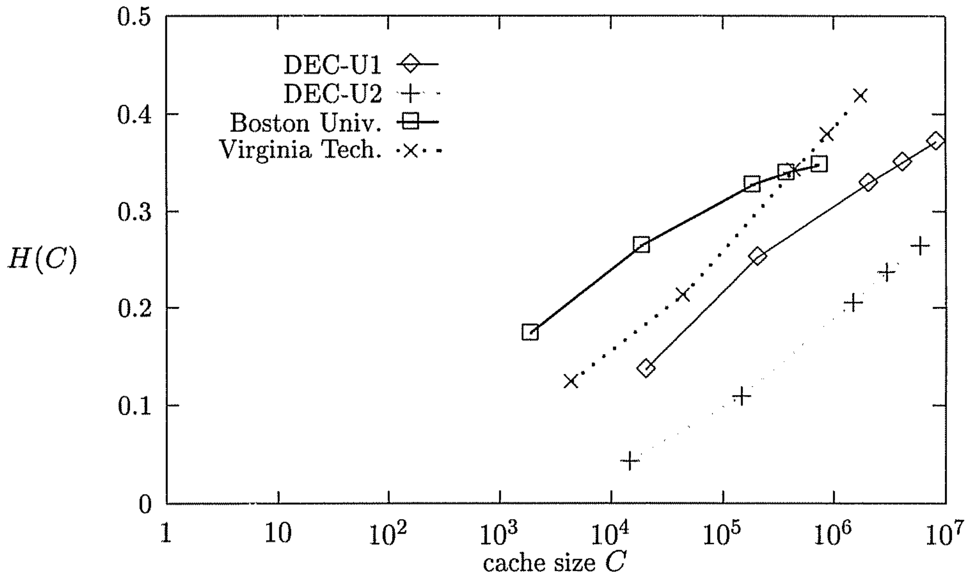

Figure 2: Hit-ratio as a function of cache size.

However, the curves do not exactly start from the origin. In other words, if we compare these curves to straight lines running through the coordinate $(1, 0)$, it would appear that the model is not particularly accurate. There are a number of reasons for this discrepancy. First, the model does not take into account document modifications at all, while the simulations that generate these hit ratios always treat modified documents as cache misses. It is known that document modifications tend to reduce Web cache hit ratio significantly [KLM97]. Thus, the model will predict a cache hit ratio that is much higher than in practice. We are working on incorporating document changes in the model. Second, the "cache size" C in the above figure is not accurate. The simulations were run assuming a cache that has a fixed capacity in terms of bytes, while the Web pages have different sizes. Thus, C in the figure is simply an estimate of the average number of documents in cache. We are planning to run more simulations assuming caches that hold a fixed number of documents to further verify the model.

Lastly, note that the slope of the curves are not necessarily identical because the slope of the approximation for $H(C)$ is dependent on N. The traces were taken during different time periods (Nov. 1994 for Boston University traces, Feb. 1995 for Virginia Tech traces, and Aug. 1996 for DEC traces), and N probably increases with time.

5 Page Request Interarrival Times

We now investigate the distribution of times between requests for a given page and show that the probability that a page will referenced k requests after it was last referenced is proportional to $1/k$.

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

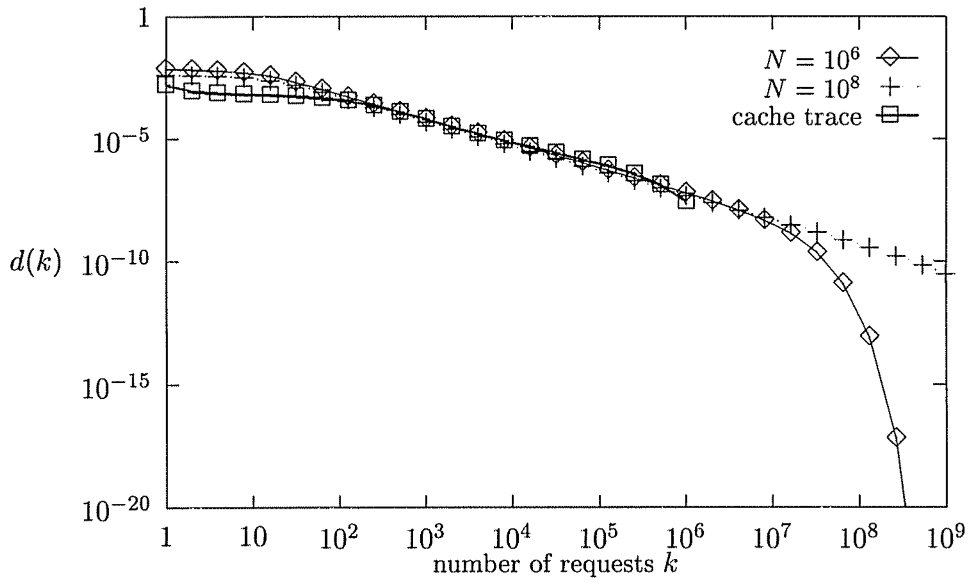

Figure 3: Probability distribution for page request interarrival times.

request for page i). Assuming that page requests are independent, we find that

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

Inserting the Zipf law gives us:

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

Noting that $\Omega \approx \frac{1}{\ln N}$ we have:

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

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

Figure 3 shows a plot of the probability distribution for page request interarrival times $d(k)$ produced by our model and the distribution for a *cache trace* of actual requests for web pages. The *cache trace* was extracted from the trace file for a DEC Web server that serviced roughly 1700 workstation over a period of 25 days [DEC96]. Once again the model predicted by Zipf's law is consistent with data observed at operational web servers [RV98, CI97], that is, the probability that a document will be referenced k requests after it was last referenced is proportional to $1/k$. Note that our model predicts higher values for $d(k)$ when $k < 10^2$ but is quite accurate when $k > 10^2$.

6 Revisiting Cache Replacement Algorithms

The model that we have discussed so far is called an *independent reference model* [GCD73] in the early operating system paging studies [Den80]. It is well known in the operating system caching community

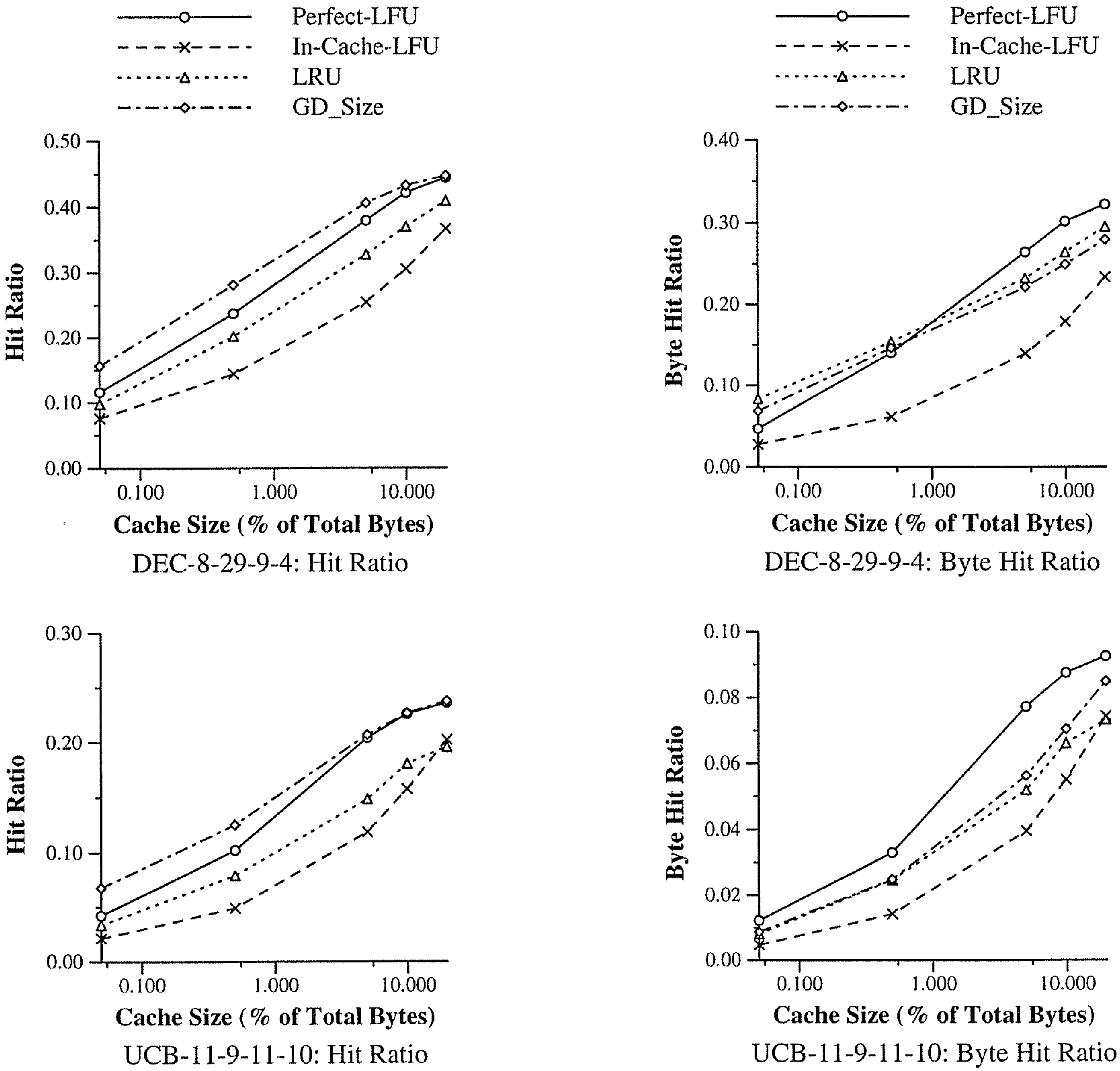

Figure 4: Hit-ratio and byte hit-ratio for four algorithms for a one-week portion of DEC traces and a two-day portion of University of California at Berkeley traces.

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 [GCD73]. In practice, an online algorithm detects those documents by keeping track of the number of references to each document and keeping 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. However, Perfect-LFU is a better realization of the optimal cache replacement algorithm under the model.

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 [CI97]. We answer the above question using trace-driven simulations.

Figure 4 shows the hit-ratios and byte hit ratios for the four algorithms—In-Cache-LFU, Perfect-LFU, LRU, and GD-Size—under a one-week portion of DEC traces [DEC96, 8/29/96 to 9/4/96] and a two-day portion of University of California at Berkeley (UCB) traces [UCB96, 9/9/96 to 9/10/96]. Although we found that other traces produced similar results, we present only the results from these two traces.

Figure 4 show 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 takes document size into account (that is, small documents are favored over large ones) whereas the Perfect-LFU does not. Note that when cache sizes are large, Perfect-LFU performs comparatively 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. We are currently designing an algorithm that has a bounded space requirement and considers both reference counts and document size.

7 Drifts in Document Hot Sets

So far we have assumed that the page request distribution is stationary, that is, the distribution does not change with time. In practice this assumption is an over-simplification because the probabilities of reference for some documents decrease while others increase. The model indicates that a small percentage of documents (the hot set) will be responsible for a large percentage of Web requests. In this section we study how the hot set in the request stream changes with time by analyzing the proxy traces.

We took the portion of DEC traces and looked at the most popular 600 URLs in each day. The one-week DEC trace from September 16 to September 22, 1996, has 4.5 million requests. In each day,

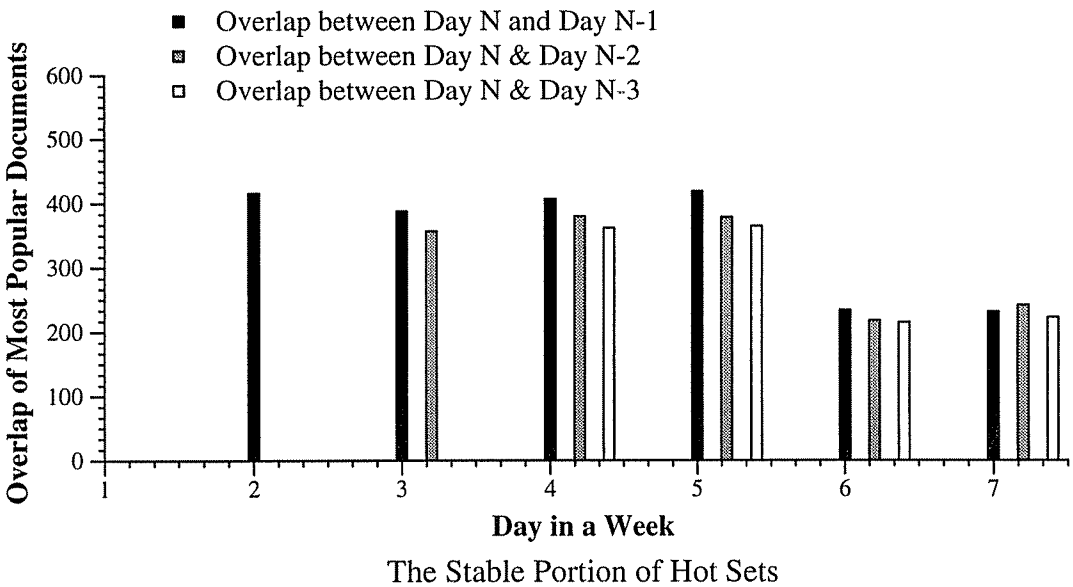

Figure 5: Overlap of hot sets for consecutive days in DEC traces.

the most popular 600 URLs accounts for over 10% of the total requests. After obtaining the hottest 600 URLs for each day, we then determined the number of documents that remain in the hot set for each consecutive day. In other words, for each day N, we looked at the size of the intersection between the hot set during day N and those for days $N-1$, $N-2$ and $N-3$. The results are shown as three bars in Figure 5.

The figure shows that about two thirds of the hot set remain unchanged over time. The exception is September 21 and September 22, which are a Saturday and a Sunday. The hot sets change by more than half compared to those of the working days. Looking at the intersection of hot sets during day N and day $N-2$, as well as day N and day $N-3$, we see that the overlap of hot documents decreases as the time interval increases. However, even after three days, about 60% of popular documents continue to be requested during the working days and about 30% of the popular documents remain popular on weekends. Thus, a significant portion of the hot documents appear quite stable.

We are still studying other traces and investigating how their hot set changes with time. We are also trying to understand the implications of hot set drifts to Web proxies, and in particular, the implication to prefetching and multicast delivery of Web pages.

8 Discussion

In this paper we have shown that a simple model based on Zipf's law can explain the asymptotic behavior for three properties that are observed in real Web cache traces. Therefore, we argue that Zipf's law should perhaps be considered as an appropriate model for Web reference streams at a particular level of abstraction. Our results also indicate that these properties found in many studies are perhaps inherent to Web access streams and not an artifact of the traces studied.

We also revisited the issue of Web cache replacement in light of the new model and found that the Least-Frequently-Used (LFU) algorithm as dictated by the new model indeed performs best in terms of byte hit-ratios. However, practical implementations of LFU, which typically keep reference counters for in-cache documents only, performs poorly and should not be used as a cache replacement algorithm.

In order to gain an understanding of how document reference probabilities change with time, we investigated the daily changes of the hot set in Web reference streams. Looking at one proxy traces, we found that about a third of the documents in the hot set (those that account for 10% of the total reference) changes from day to day. On the other hand, about 60% of the documents remain in the hot set for more than three days.

Although we have argued in this paper that Zipf's law enables us to construct a reasonably accurate model, some researchers believe that Zipf's law is not accurate enough. 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 [ABCdO96].

Clearly our model has limitations. First, the model does not consider the cache's replacement policy, which no doubt plays a critical role in a cache's performance. Second, Zipf's law over-constrains the request probability function as $P(i) = i^{\alpha}$ where $\alpha = -1$. One question that should be investigated is: how would the results for hit-ratio and interarrival times change if the exponent α were not constrained? Third, the model does not consider document modifications. However, our mathematical model has the advantage that it can predict the *asymptotic* behavior of three properties where as regular simulations would require overly large trace files to obtain such asymptotic results.

Our goal is to find a model that is accurate enough to assist Web cache design and configuration. We are currently trying to improve the model and trying to understand its applicability to other proxy traces as well as its implications for Web cache design. One interesting question is whether the Zipf's model applies to Web accesses seen by the parent proxy in a proxy hierarchy.

References

[ABCdO96] 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.

[CBC95] 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.

[CI97] 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.

[DEC96] Digital Equipment Corporation, Digital's web proxy traces, 1996. ftp://ftp.digital.com/pub/DEC/traces/proxy/ webtraces.html.

[Den80] Peter J. Denning. Working sets past and present. *IEEE Transaction on Software Engineering*, SE-6(1):64–84, January 1980.

[DMF97] Bradley M. Duska, David Marwood, and Michael J. Feeley. The measured access characteristics of world-wide-web client pr oxy caches. In *Proceedings of USENIX Symposium on Internet Technology and Systems*, December 1997.

[GB97] 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/.

[GCD73] Edward G. Coffman, Jr. and Peter J. Denning. *Operating Systems Theory.* Prentice-Hall, Inc., 1973.

[Gla94] 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.

[KLM97] 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.

[RV98] 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.

[UCB96] UCB home IP HTTP traces, 1996. http://www.cs.berkeley.edu/ gribble/traces/index.html.

[WAS+96] 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.