EFFECTIVE CACHING OF WEB OBJECTS USING ZIPF'S LAW

D.N. Serpanos

Dept. of Computer Science
University of Crete
Heraklion, Crete
Greece

G. Karakostas

Dept. of Computer Science
Princeton University
Princeton, NJ 08544
USA

W.H. Wolf

Dept. of Electrical Eng.
Princeton University
Princeton, NJ 08544
USA

ABSTRACT

Web accesses follow Zipf's law with a good approximation, as measurements and observations indicate. This property provides an important tool in the design of web caching architectures, because it allows designers to calculate appropriate cache sizes to achieve the desired hit ratios. The appropriate cache size combined with an LFU replacement policy achieves high cache hit rates. However, LFU replaces objects based on frequency measurements of past accesses. Thus, the system achieves high hit rates only after these measurements are reliable and converge to the final Zipf distribution. In this paper, we provide an analysis using Chernoff's bound and a calculation of an upper bound of the number of initial requests that need to be processed in order to obtain measurements of popularity with high confidence and a measured Zipf distribution which converges to the correct one.

1. INTRODUCTION

This paper describes new results that help us more efficiently cache Web objects. Efficient caching is particularly important for multimedia data on the Web—because audio and video objects are large, they present different challenges for Web caching than do simple text pages. Zipf's law helps us select which objects to cache, but to use it we must be able to experimentally measure the popularity of cacheable objects in our system. This paper describes the conditions under which we can reliably apply Zipf's law to a collection of objects.

Traditional Web caching methods assume that the objects are small, such as the typical HTML page, which is often less than 1 KB in size. For example, the Harvest/Squid system [2] is the best-known Web caching system. It is organized as a hierarchy of proxy caches, with requests to uncached objects being passed up the hierarchy as far as necessary to retrieve the requested object. When the object is retrieved, it is cached at every intermediate proxy cache between the source and destination—this is often called a *cache-everywhere* policy. Cache-everywhere makes sense when the objects are relatively small.

However, many multimedia objects are relatively large. As a result, excessive caching burns up two important resources: cache disk space and bandwidth. Video files are sufficiently large that caching even a moderate amount of video can fill up most practical caches. Furthermore, transmitting large files consumes bandwidth that might be better used if another object that is more likely to be requested could be transmitted in its place. Kozuch and Wolf [5] developed algorithms and heurisitcs for placing large multimedia objects in a hierarchical cache structure to optimize the usage of disk space and bandwidth; however, this work did not consider how to choose which objects should be cached.

One rule for choosing which objects to cache that is receiving increasing attention is Zipf's law, which has been applied in a number of disciplines. Zipf's law predicts that the probability of access for an object is a function of its popularity: the n^{th} most popular object will be accessed with a probability proportional to $1/n$. (Because this series does not sum to 1, the $1/n$ factor must be weighted to create a probability.) Bestavros et al [4] showed that Web accesses can be modeled by Zipf's law. Serpanos and Wolf [6] calculated the size of a cache that is required to achieve a desired hit ratio using Zipf's law caching. This work showed that high hit rates can be achieved using Zipf's law caching. This work was independently verified by Breslau *et al.* [3].

Zipf's law, however, does not directly directly tell us how to select which objects to cache. In a realistic Web caching system, we do not have *a priori* knowledge of document popularities; we must instead deduce the relative importance of documents by observing Web traffic. In this paper, we derive new **confidence bounds** for determining the Zipf's law distribution for a Web cache. Our results give the number of observations of Web requests that must be observed to obtain a Zipf's law ranking of objects with a desired confidence. For simplicity, we concentrate on the single Web cache case site in the Web, as indicated in Fig-

ure 1. In this model, one site, which contains many users is attached to a single cache (gateway) which serves the whole population of users (clients). The cache is receiving data (objects) potentially from all servers attached to the Internet.

In Section 2 we describe the system environment we consider and introduce the notation used throughout the paper. In Section 3 we describe the Zipf distribution of the clients' requests and calculate the sizes of the caches required to achieve high cache hit rates. In Section 4, we calculate the bound on the number of client requests that is necessary for the cache to accumulate before it considers the Zipf distribution "reliable" for use for LFU.

2. MODEL AND NOTATION

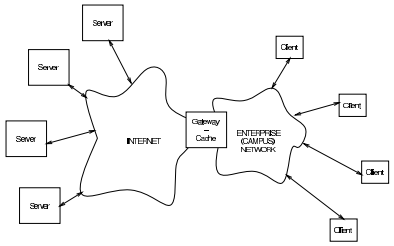

Figure 1: Network Model

Figure 1 shows a typical configuration of a site, which provides a cache in the gateway between the enterprise (or campus) network and the Internet. In this model, clients make requests for objects that reside on any of the servers shown in the figure; we denote the sequence of requests from the clients with R. The gateway, which caches objects, serves a request if the requested object is stored in cache, otherwise it forwards the request to the appropriate server.

Considering the observations that Internet accesses (object requests) follow a Zipf (or Zipf-like) distribution, we assume that the stream of client requests R is a series of *independent* trials drawn from a Zipf (or Zipf-like) distribution over a set of N possible items (web pages or sites, in our case). Specifically, we use the following assumptions and notation:

1. there is a set S of N objects, $S = \{O_i \mid 1 \leq i \leq N\}$, which will be accessed by a group of users during a time interval t_I;

2. there is a known popularity of the N objects, i.e. which one it the most popular, the 2-nd most popular, etc., and that the index i in the notation O_i indicates this popularity (the Zipf rank).

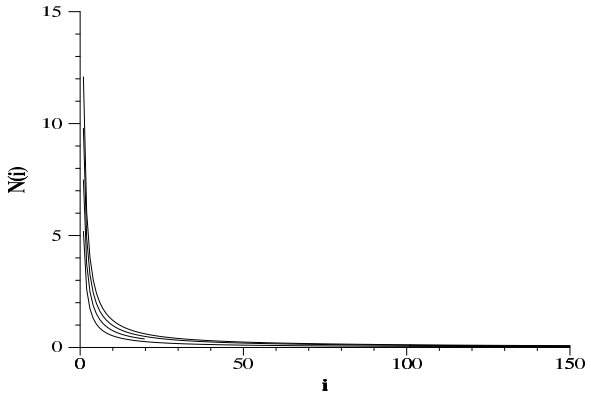

Figure 2: Zipf's function

3. ZIPF'S LAW AND CACHING

Given the set of N objects used by a set of clients, Zipf's law allows one to calculate the number of accesses (uses) to each object based on its popularity. Specifically, Zipf's function quantifies the probability that an access is made to object O_i: $P_i = \frac{a}{i}$, where a is a constant. Constant a is easily calculated, since the sum of all probabilites is equal to 1: $\sum_{i=1}^{N} P_i = 1 \Rightarrow a \times H_N = 1 \Rightarrow a = \frac{1}{H_N} \approx \frac{1}{\ln N}$, where H_N is the N-th harmonic number, which is approximated with $\ln N$. Thus,

$$P_i = \frac{a}{i} = \frac{1}{H_N \times i} \tag{1}$$

Figure 2 plots the probabilities for $N = 10^i$, where $2 \leq i \leq 5$.

Then, one can calculate the number of accesses of the k most popular objects O_1, O_2, ..., O_k as follows. If a number of accesses N_A is directed to the set of the N objects and N_A is large enough, then, in general, object O_i, $1 \leq i \leq N$, will be accessed $P_i \times N_A$ times, based on Zipf's law. So, the total number of accesses to the k most popular objects is:

$$\sum_{i=1}^{k} N_A \times P_i = N_A \times \sum_{i=1}^{k} P_i = N_A \times \frac{H_k}{H_N} \tag{2}$$

This implies that, if we have a "hot" cache that serves the requests, which stores *only* the k most popular objects, then the cache hit rate is:

$$h = \frac{N_A \frac{H_k}{H_N}}{N_A} = \frac{H_k}{H_N} \tag{3}$$

Based on the above, we can calculate k, the number of objects in the cache, which can achieve a given hit-ratio h, from Equation 3:

$$H_k = h \times H_N \Rightarrow \ln k = h \times H_N \Rightarrow k = e^{h \times H_N} \tag{4}$$

The calculations indicate that the given cache hit ratio h will be observed (measured) under the following two conditions:

1. Zipf's law holds for the set of accesses and objects measured;
2. the time interval during which measurements are made is large enough.

Note that, one can certainly develop scenarios where the accesses are in such an order that the cache hit rate becomes significantly lower than the expected h for short time intervals (e.g., when many consecutive accesses are targeted to the least popular objects). However, the cache hit ratio is the expected h for long enough intervals, i.e. long enough request streams. In the following, we calculate an upper bound for the minimum length of the request stream, so that the measured object popularities are reliable (with high confidence) and the Zipf distribution estimated up at that point converges to the final one.

4. THE CONFIDENCE BOUND

Considering that the request stream R is a series of *independent* trials drawn from a Zipf distribution over the set S of N possible objects, at any point in R, the next request will be the i-th most popular of the N objects with probability $P(i) = \frac{a}{i}$, where $a = \frac{1}{H_N} \approx \frac{1}{\ln N}$. For the purposes of our analysis, we consider the environment *closed*, i.e. that the set S of the N objects does not change (none of the objects "dies" or changes, and no new objects are born).

In order to perform our analysis, we introduce the concept of a *past*, P, of a stream request R: P is a prefix of R. We define as $n_P(i)$ the number of appearances in P of the i-th most popular object (in R). It follows that the expected value of $n_P(i)$ is:

$$E[n_P(i)] = \frac{\mid P \mid}{i \ln N} \tag{5}$$

where $\mid P \mid$ is the length of P. For simplicity, we denote this value as $E(i)$ in the following.

Given the concept of a *past* in a request stream, the problem we solve is the following: given a random R, how long should the past P be, so that the access frequencies in P render reliable measurements of object popularities that reflect exactly the distribution of the N objects for the *entire* R with very high probability?

The answer to this question provides information about the convergence of P to the real (final) Zipf distribution. We can provide theoretical upper bounds by taking advantage of the *knowledge* of the distribution in R and the assumption of *independence* between the requests in R. These assumptions are very strong, but experimental results support their validity [3].

In order to quantify the concept of confidence described above, we introduce the metric of *difference*, $D(i)$, for every object O_i:

$$D(i) = E(i) - E(i+1) \stackrel{(5)}{=} \frac{\mid P \mid}{i(i+1)} \tag{6}$$

Using $D(i)$, we characterize a past P as a *good past*, as follows.

Definition 4.1 (Good past) *A past P of a random stream of requests R is a* **good past** *of R if the following condition is met:*

> *$n_P(i)$, the number of appearances of O_i in P, is within distance $\frac{D(i)}{2}$ of its expected value, i.e. the following holds:*
>
> $$|n_P(i) - E(i)| \leq \frac{D(i)}{2}, \;\; i = 1, 2, \ldots, N \tag{7}$$

If condition (7) holds for all objects, then the objects have exactly the same popularity ordering in P as they have in R. This can be easily deduced:

$$\left.\begin{array}{rcl} -\frac{D(i)}{2} & \leq & n_P(i) - E(i) \\ -\frac{D(i+1)}{2} & \leq & E(i+1) - n_P(i+1) \end{array}\right\} \stackrel{(+)}{\Rightarrow}$$

$$\Rightarrow n_P(i) - n_P(i+1) \geq -D(i) - \frac{D(i) + D(i+1)}{2} \stackrel{(6)}{>} 0$$

for all i.

Effectively, the definitions of *difference* and *good past* allow us to specify a confidence radius around the expected value of each O_i in such a way so that, if the number of appearances falls into their confidence intervals, the objects retain their ordering in R (which is the same as the ordering of the $E(i)$'s), because the confidence intervals do *not* intersect. Based on the above, our problem becomes: how long should P be, so that it is a good past with very high probability?

Theorem 4.1 *For any $\epsilon > 0$, a past P of R of length $2N^2(N+1)^2 \ln^2 N \ln \frac{2N}{\epsilon}$ is a good past with probability at least $1 - \epsilon$.*

Proof: In our analysis we use the following Chernoff bound [1]:

Lemma 4.1 (Chernoff bound) *Let $X_1, X_2, \ldots, X_n$ be mutually independent random variables such that*

$$Pr[X_i = 1] = p$$

$$Pr[X_i = 0] = 1 - p$$

for some $p \in [0, 1]$. Let $X = X_1 + X_2 + \ldots + X_n$ and $E[X] = pn$. Then

$$Pr[|X - pn| > \theta] \leq 2e^{-\frac{2\theta^2}{n}} \tag{8}$$

for any $\theta > 0$.

We define the following sequence of random variables for each O_i:

$$w_j(i) = \begin{cases} 1, & \text{if } j\text{-th request of } R \text{ is } O_i \\ 0, & \text{otherwise} \end{cases}, \; j = 1, \ldots, W$$

Then given that $w_j(i)$'s are mutually independent for all j, $n_W(i) = \sum_{j=1}^{W} w_j(i)$ and $Pr[w_j(i) = 1] \approx \frac{1}{i \ln N}$, due to Zipf's function. Thus, we can apply Lemma 8 with $p = \frac{1}{i \ln N}$ and $\theta = \frac{D(i)}{2}$, obtaining

$$\begin{aligned} Pr[|n_P(i) - E(i)| > \frac{D(i)}{2}] &< 2e^{-\frac{P}{2i^2(i+1)^2 \ln N}} \\ &\leq 2e^{-\frac{P}{2N^2(N+1)^2 \ln^2 N}} \quad (9) \end{aligned}$$

Note that

$$\begin{aligned} Pr[P \text{ is } \textit{not} \text{ good past}] = \; & Pr[\text{ (7) not true for } O_1 \\ & \vee \text{ (7) not true for } O_2 \vee \ldots] \\ \leq \; & \textstyle\sum_{i=1}^{N} Pr[\text{ (7) not true for } O_i] \\ \overset{(9)}{<} \; & 2Ne^{-\frac{P}{2N^2(N+1)^2 \ln^2 N}} \quad (10) \end{aligned}$$

If our 'confidence' parameter is ϵ with $0 < \epsilon \leq 1$, then it must be true that

$$Pr[P \text{ is } \textit{not} \text{ a good past}] < \epsilon$$

But then from (10) we get that we must pick P so that

$$P \geq 2N^2(N+1)^2 \ln^2 N \ln \frac{2N}{\epsilon}$$

□

Notice that this bound is relatively large in terms of N, especially under the assumption that during this period the system is closed. The size of the bound is due to the very strong condition we want to satisfy (condition (7)) to obtain a good past.

5. CONCLUSIONS

This paper has introduced new results on the experimental determination of a Zipf's law distribution for a set of objects based upon a stream of requests for those objects. Our results allow a Web cache that takes advantage of Zipf's law to accurately determine the usage distribution of the objects it should cache based upon observed usage patterns for the objects.

Efficient caching is very important to the development of Web-based multimedia applications. Multimedia objects are sufficiently large that they can consume unacceptable amounts of both disk space and bandwidth when cached indiscriminately. Zipf's law holds the promise of more effective use of network caching resources for multimedia objects. We are presently constructing an experimental caching system based on Zipf's law. Advanced multimedia caches can help with the distribution of both streaming and non-streaming multimedia objects. We believe that the effective use of Zipf's law will be an import component of next-generation multimedia caching systems.

6. REFERENCES

[1] N. Alon, J.H. Spencer, and P. Erdos. *The Probabilistic Method*. John Wiley and Sons, 1992.

[2] C.M. Bowman, P.B. Dantzig, D.R. Hardy, U. Manber, and M.F. Schwartz. The Harvest Information Discovery and Access System. *Computer Networks and ISDN Systems*, 28:119–125, 1995.

[3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *In Proceedings of IEEE INFOCOM'99, New York, NY, USA*, March 21-25 1999.

[4] C.R. Cunha, A. Bestavros, and M.E. Crovella. Characteristics of WWW Client-based Traces. Technical Report BU-CS-95-010, Computer Science Department, Boston University, July 1995.

[5] M. Kozuch, W. Wolf, and A. Wolfe. An Approach to Network Caching for Multimedia Objects. In *Proceedings, ICCD '97*. IEEE Computer Society Press, 1997.

[6] D.N. Serpanos and W. Wolf. Caching Web Objects using Zipf's Law. In *Multimedia Storage and Retrieval Systems*. SPIE, 1998.