Analysis of Web Workloads Using the Bootstrap Methodology

Johnson Lee William Miniscalco Meng Li W. David Shambroom John Buford

Verizon Communications
40 Sylvan Road
Waltham, MA 02451

ABSTRACT

Modeling the performance of caches requires examining their behavior when subjected to realistic workloads that are created by extracting the essential characteristics of actual Web traffic. We have used the Bootstrap methodology for analyzing response size and file popularity distributions obtained from proxy log files. The Bootstrap is a resampling technique that enables one to estimate both population parameters and their standard errors when no accurate mathematical representations are available for the underlying distributions. For this investigation we used real-world workloads from NLANR and Verizon Laboratories together with a synthetic workload generated by Web Polygraph. The analysis of response size included averages and percentiles, processing and bandwidth costs, and a null hypothesis test for ascertaining the likelihood that different datasets have the same underlying distribution. The file popularity distribution was investigated to assess the degree of Zipf-like behavior. We conclude that the Bootstrap is an effective tool for the analysis of nonparametric distributions that enables one to determine confidence levels and make stronger statements than possible using conventional statistics. It is also valuable for determining how accurately a synthetic workload represents actual traffic.

KEYWORDS

Web caching; Proxy cache; Web workloads; Bootstrap

1. INTRODUCTION

An effective implementation of Web caching requires a detailed understanding of the temporal and spatial characteristics of the traffic load together with their impact on cache performance. A network service or content distribution network (CDN) provider with a national or international footprint needs to deploy a widely distributed network of caching proxies to provide the required quality of services. Issues that must be resolved include sizing the caches, loadbalancing of cache clusters, distributing traffic between caches sites, backup/fail-over policies, and managing performance for service level agreements. The behavior of a distributed system with many clients, servers, and networks depends heavily on the characteristics of its workload [2,6,8,12,13,14,16]. Thus, one of the first steps in any performance evaluation study is to understand the workload by extracting key characteristics from actual Web workloads.

In order to analyze distributions of important parameters such as file size from workload sources or evaluate the accuracy of a synthetic workload, it is extremely valuable to use a nonparametric statistical method (*i.e.*, one without an assumed mathematical expression for the distribution function). Adequately understanding a distribution normally requires knowledge of more than the mean. In the absence of a mathematical expression for the distribution, however, evaluating statistics poses a problem because for most estimators other than the mean there are no procedures for obtaining the uncertainty of the estimate. In this work we adopt the Bootstrap methodology [9,10] to perform statistical estimates and inferences of Web workload parameters that include estimates of their errors.

The objective of this paper is to use the Bootstrap methodology to investigate the variations of workloads obtained from different sources by comparing their response sizes and popularity distributions. This includes the evaluation of standard errors and the confidence with which a synthetic workload represents measured data. The remainder of this paper is organized as follows: In Section 2 a brief description of the Bootstrap methods is provided. In Section 3 the data collection procedures are discussed. In Section 4 are presented statistical estimates and inferences determined by applying the Bootstrap methods to the datasets. We focus our attention on the response size distributions and the distributions for the file popularity. Finally, a summary is presented in Section 5. In the Appendices the Bootstrap algorithms relevant to this paper are described.

2. THE BOOTSTRAP METHODOLOGY

2.1 Motivation

Since the Bootstrap methodology was introduced in 1979 by Efron [9,10], it has been widely used for statistical problems in a variety of fields [7,11]. The Bootstrap is a computer-based method of inference that can answer many important statistical questions without assuming a particular functional form for the underlying probability density function. One of the goals of the Bootstrap is to provide a mechanism for evaluating inferences such as the variance, the confidence interval, and the significance level.

Many of the distributions encountered in the analysis of Web traffic have no *a priori* functional form. If it is necessary to represent them by mathematical expression (*e.g.,* to generate synthetic workloads for experiments or tests), it is typically done through piece-wise fits with common distribution functions [2] or using a superposition of such functions [18]. Statistics and error estimates calculated using such fits are generally valid only for comparison to the original dataset to assess goodness of fit. Thus evaluating statistics on measured datasets, whether for direct analysis or evaluating a fit, must be done nonparametrically. The explicit recognition of uncertainty is central to statistical analysis, and any statistical estimates that do not include confidence levels or estimated standard errors are of limited utility. However, for nonparametric distributions there are in general no procedures for evaluating the error of an estimate beyond those for the error of a mean. The Bootstrap and some other computer-based techniques, such as permutation tests and nonparametric inferences, play an important role in resolving such difficulties.

The essence of the Bootstrap is to create additional samples of the population by resampling the original dataset with replacement (*i.e.,* allowing a specific datum to be selected more than once). A wide range of statistical problems can be addressed this way, eliminating the need to oversimplify complex problems. The reader is referred to the Bootstrap literature for details of the method [7,9,10,11].

2.2 Estimating Values Using the Bootstrap

In this section we illustrate how the Bootstrap is used to estimate values. Appendix A describes how it can be applied to determining the standard errors of these estimates. Let *x'* be a continuous random variable with values *x* whose probability density function (pdf) is *f(x)*. The cumulative distribution function (CDF) of *f(x)* is *F(x)* and is defined in the usual way as

$$\mathrm{F(x) = P(x' \le x)} = \int_0^x \mathrm{f(x')dx'} \quad , \tag{1}$$

$$\text{with} \qquad \mathrm{F}(\infty) = 1 \quad , \tag{2}$$

where *P* is the probability of selecting a value of *x'* in the range from 0 to *x*. The complimentary function (CCDF) of *F(x)* is $\underline{F}(x)$ and is given by

$$\underline{\mathrm{F}}(\mathrm{x}) = 1 - \mathrm{F(x)} \quad . \tag{3}$$

Suppose that a random sample **x**=(x_1, x_2,…, x_n) of size *n* is drawn from a population whose pdf and CDF are denoted by *f* and *F*, respectively. If all *n* observations x_i are identically distributed and mutually independent of each other, then all of the x_i are described by the same CDF. For the current problem x_i represents a datum (such as response size) from the log file record *i*. When there is no mathematical model for the CDF, the statistical analysis is nonparametric and uses only the fact that the random variables are independent and identically distributed (iid). An important role is played in nonparametric analysis by the empirical distribution that assigns equal probability *1/n* to each sample value x_j. The empirical distribution function (EDF), $\hat{F}$, is the corresponding estimate of *F*, which is defined as the sample proportion

$$\hat{\mathrm{F}}(\mathrm{x}) = \#\{\mathrm{x_j} \le \mathrm{x}\} / \mathrm{n} \quad , \tag{4}$$

where #{A} means the number of times the event *A* occurs in the sample. More explicitly, we may write

$$\hat{\mathrm{F}}(\mathrm{x}) = \frac{1}{\mathrm{n}}\sum_{j=1}^{n}\mathrm{H(x - x_j)} \quad , \tag{5}$$

where $H(x\text{-}x_j)$ is the Heaviside (unit step) function which jumps from 0 to 1 at x= x_j. Notice that the values of the EDF are discrete (*0*, *1/n*, *2/n*,…, *n/n*), so the EDF is equivalent to its points of increase, the ordered values $x_{(1)} \le x_{(2)} \le \cdots \le x_{(n)}$. The EDF plays the role of fitted model when no mathematical form is assumed for *F*.

The sample (dataset) ***x*** is used to make inferences about a population characteristic, generically denoted by *q*, using a statistic whose value in the sample is *t*:

$$\theta = \mathrm{t(F)} \quad , \tag{6}$$

where *t(F)* is a statistical function that specifies how to compute *t* from *F*. Because *F* is unknown we estimate *q* by substituting the EDF for *F*:

$$\hat{\theta} = \mathrm{t}(\hat{\mathrm{F}}) \quad . \tag{7}$$

The Bootstrap methodology relies upon the above substitution. In Bootstrap terminology this is referred to as the Plug-in Principle. Simple examples of statistics are the mean and the variance of the population, which are defined, respectively, as

$$\mathrm{t(F)} = \mu = \int \mathrm{u\ dF(u)}, \quad \mathrm{t(F)} = \sigma^2 = \int \mathrm{u^2\ dF(u)} - [\int \mathrm{u\ dF(u)}]^2 \ . \tag{8}$$

By applying Eq.(5) and Eq.(7) to estimate the sample mean, $\overline{\mathrm{X}}$, we obtain

$$\hat{\theta} = \overline{\mathrm{X}} = \mathrm{t}(\hat{\mathrm{F}}) = \int \mathrm{u\ d}\hat{\mathrm{F}}(\mathrm{u}) = \frac{1}{\mathrm{n}}\sum_{j=1}^{n} x_j \tag{9}$$

and, similarly, for the sample variance $\bar{\mathrm{s}}^2$, we have

$$\hat{\theta} = \bar{\mathrm{s}}^2 = \frac{1}{\mathrm{n}}\sum_{j=1}^{n} x_j^2 - [\frac{1}{n}\sum_{j=1}^{n} x_j\ ]^2 \ . \tag{10}$$

It can be shown that μ is equal to $\overline{\mathrm{X}}$ and t($\hat{\mathrm{F}}$) is unbiased, while σ^2 is not equal to $\bar{\mathrm{s}}^2$ and t($\hat{\mathrm{F}}$) is biased by σ^2 = n $\bar{\mathrm{s}}^2$/(n-1). The bias of the sample variance is relatively unimportant in this context because of the large ($n\sim10^6$) sample size used here. The Bootstrap algorithm for estimating standard errors is presented in the Appendix A. Several authors have suggested implementing Bootstrap programs using the statistical programming languages S and S-plus [7,11]. We implemented the Bootstrap algorithms in both Fortran with IMSL subroutines [17] and S-plus. We

Table 1. Statistical Estimates and Conclusions

Characteristic	NLANR-bo1	VZL	POLY
(1) Mean File Size (bytes) B*	8,080 (±200)	8,398 (±69)	11,027 (±49)
(2) Mean File Size (bytes) CS‡	8,100 (±200)	8,409 (±67)	11,029 (±48)
(3) Median File Size (bytes) B*	1,170.1 (±0.3)	1,407 (±3)	5,235 (±11)
(4) Median File Size (bytes) CS‡	1,170	1,407	5,234
(5) 99th Percentile File Size (bytes) B*	63,660 (±300)	83,880 (±620)	51,120 (±230)
(6) 99th Percentile File Size (bytes) CS‡	63,700	83,860	51,110
(7) Maximum File Size (bytes)	1.9 x 10^8	1.6 x 10^7	3.4 x 10^6
(8) Bootstrap H_0: F=G	ASL=48% (Ref)	ASL=15%	ASL<0.01%
(9) Zipf-like Exponent	0.853	0.902	NA

* B indicates value obtained using the Bootstrap

‡ CS indicates value obtained using conventional statistics

used the Fortran versions because of a factor of 10 speed advantage.

3. DATA COLLECTION

The values of the characteristic parameters of the workload may vary from one server to another due to differences in client populations. Therefore, we extracted workloads from proxy log files obtained from three different sources:

1. National Laboratory for Applied Network Research (NLANR-bo1) hierarchical cache system, where bo indicates the NCAR site in Boulder, CO [15],
2. Verizon Laboratories' (VZL) proxy server without caching capabilities [4], and
3. A Lucent caching proxy subjected to a load generated by the Web Polygraph benchmarking system running the Polymix-3 workload (POLY) [18].

Log files obtained from NLANR-bo1 and VZL represent real-world workloads while those obtained from POLY are synthetic workloads. A test system such as Polygraph does not exactly reproduce an actual Web workload for reasons of computational efficiency and the need to reproduce long-term cache behavior within a restricted test schedule [19]. Accordingly, it is interesting to anlayze the differences between actual and synthetic workloads.

Log entries from all three sources included response size, elapsed time, and a time stamp. From NLANR-bo1 we used five days (September 18-22, 2000) of sanitized cache access logs (from bo1.sanitized-access.20000919.gz to bo1.sanitized-access.20000922.gz) and stored the data in a database with a total of 2.4 x10^6 records. Only weekdays were examined to exclude the effect of different behavior on weekends. Similarly, we used one day (June 5, 2000) of VZL data and stored them in a database with 1.4 x10^5 records. For the synthetic workload from POLY, we collected a dataset over a period of 160 minutes after the cache server reached steady-state. This generated 5 x 10^5 records. The full datasets were used in all calculations unless otherwise noted.

4. ANALYSIS OF WEB WORKLOADS

4.1 Distributions for the workloads

To determine a response size distribution, we queried the appropriate database and counted the number of requests (frequency) for each response size *fs*, independent of file identity. A bin size of Δ*fs*=1 byte was used and the distribution was normalized to construct the pdf as a function of *fs*, *f(fs)*. The CCDF $\underline{F}(fs)$ is determined using Eqs. (1)-(3). Figure 1 shows the variation of $\log_{10}[\underline{F}(fs)]$ with $\log_{10}[fs]$ for the NLANR-bo1, VZL and POLY workloads. It is apparent that the real-world workloads (NLANR-bo1 and VZL) behave similarly and have much longer tails than the synthetic workload (POLY). This heavy-tailed behavior has been widely noted for file and response size distributions in Web traffic. Statistical techniques for heavy-tailed distributions are discussed in [1].

4.2 Bootstrap estimates and inferences

Table 1 lists statistical estimates, with standard errors in parenthesis, for NLANR-bo1, VZL and POLY obtained using both the Bootstrap and conventional statistical methods. The latter consists of calculating the desired statistic on a single sample from a population and, except for the mean, cannot provide an estimate of error. For convenience we split the discussion of the estimates into four groups: (A) sample means and percentiles, (B) processing and bandwidth cost as a function of response size, (C) Bootstrap's null hypothesis H_0 test, and (D) file popularity distributions. For the analyses we used 500 interations of the Bootstrap resampling technicque (B=500). The parameters obtained can be used to build or validate workload models to capture the important features and characteristics of the distributions.

A. Sample Averages and Percentiles

Row (1) of Table 1 is the mean of the response sizes in bytes with the standard deviation of the sample mean in parenthesis, both calculated using the Bootstrap on the full datasets. POLY has the largest mean response size with the smallest standard deviation, while NLANR-bo1 and VZL

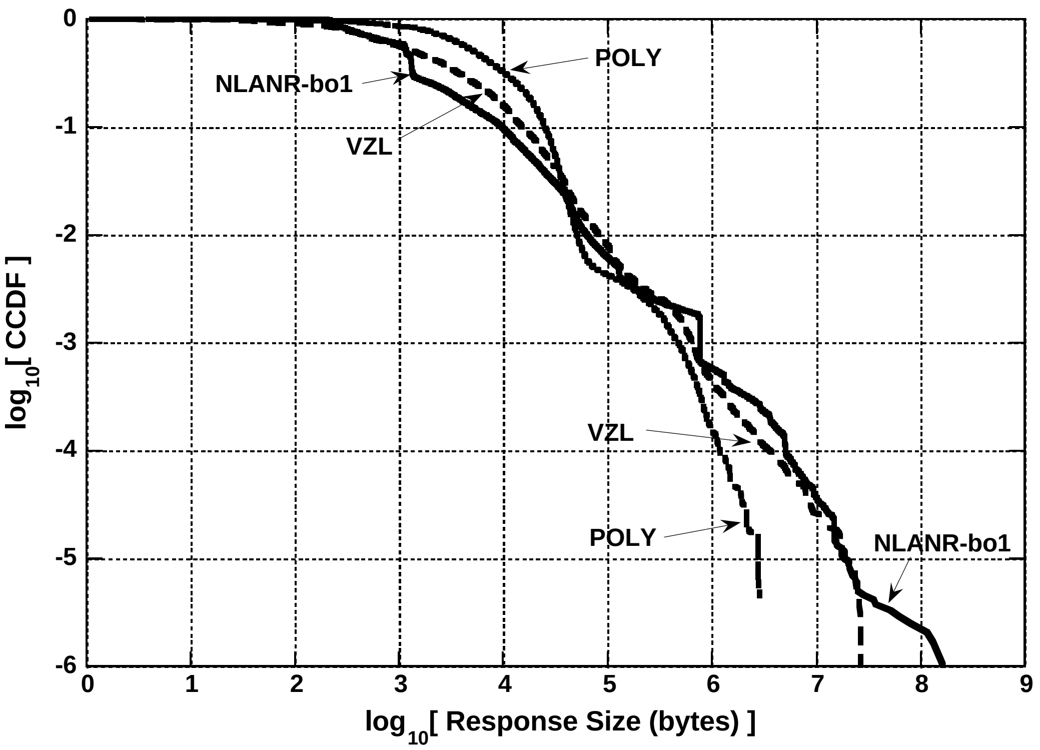

Figure 1: Complimentary cumulative distribution function versus response size for the NLANR-bo1, VZL, and POLY datasets.

have smaller mean response sizes with larger (in the case of NLANR-bo1 much larger) standard deviations because of the heavy tails seen in Figure 1. The value of mean response size for POLY agrees with the input specification used for Web Polygraph. Row (2) shows the same statistics calculated using conventional methods. The close agreement between the corresponding values in Rows (1) and (2) indicate that the Bootstrap adds little value if only the mean of a distribution is required. Rows (3) and (4) are the median response sizes calculated using Bootstrap and conventional statistics, respectively. Rows (5) and (6) are the 99th percentile response sizes determined by both methods, while Row (7) is the maximum file size for each dataset. Another indication of how skewed NLANR-bo1 and VZL response size distributions are relative to the synthetic workload is the ratio between their 99th percentile and median response sizes. For the NLANR-bo1 and VZL datasets this ratio is 54, while for POLY it is 9.7. It is important to note that while the median and 99th percentile values calculated by both methods are extremely close, only the Bootstrap provides an estimate of uncertainty.

B. Processing and Bandwidth Cost as a Function of Response Size

In assessing the impact on a caching system of a distribution of response sizes, one needs to consider both the file size and frequency with which files of that size are requested. Caching large files can save considerable network bandwidth and load on the origin server, but can consume a significant amount of disk space at the cache and cause long waits on requests for smaller files while a larger file is being processed. This trade-off analysis can be aided by computing the impact of a given file size, which can be done by taking the product of the file size and probability of a request for that file size, C(fs) = fs x Pr(fs). This provides the marginal cost for processing files of size *fs*. Here we made the simplifying assumption that the bandwidth and processing costs are proportional to the file size. We took a bin size, *Dfs*, of 100 bytes with a probability, *Pr*, of finding a file of size *fs* in the bin. Examining the shape of *C(fs)* plotted versus *fs* is particularly helpful in assessing the cost of very large files.

Statistical estimates for *C(fs)* were made using the Bootstrap with B=500 and are plotted in Figure 2(a), (b) and (c) for NLANR-bo1, VZL and POLY, respectively. Because the marginal cost is plotted on a logarithmic scale, bins containing no files cannot be displayed and the plots are cutoff at file sizes for which this becomes significant. Both the NLANR-bo1 and VZL data have no clearly defined peak but roll off from maxima at a file size in the region of 1,000 bytes. In contrast, the POLY data cover a significantly smaller range of marginal costs and have a very broad peak at file sizes between 3,200 and 20,000 bytes. The notch in the POLY data just above 3,000 bytes is reproducible and is being investigated further. We conclude that for the actual Web workloads, the marginal costs for processing files of increasing size falls off faster than linearly because the probability of requesting a file of a particular size decreases faster than the file size grows. Because the Polymix-3 workload produces a more uniform

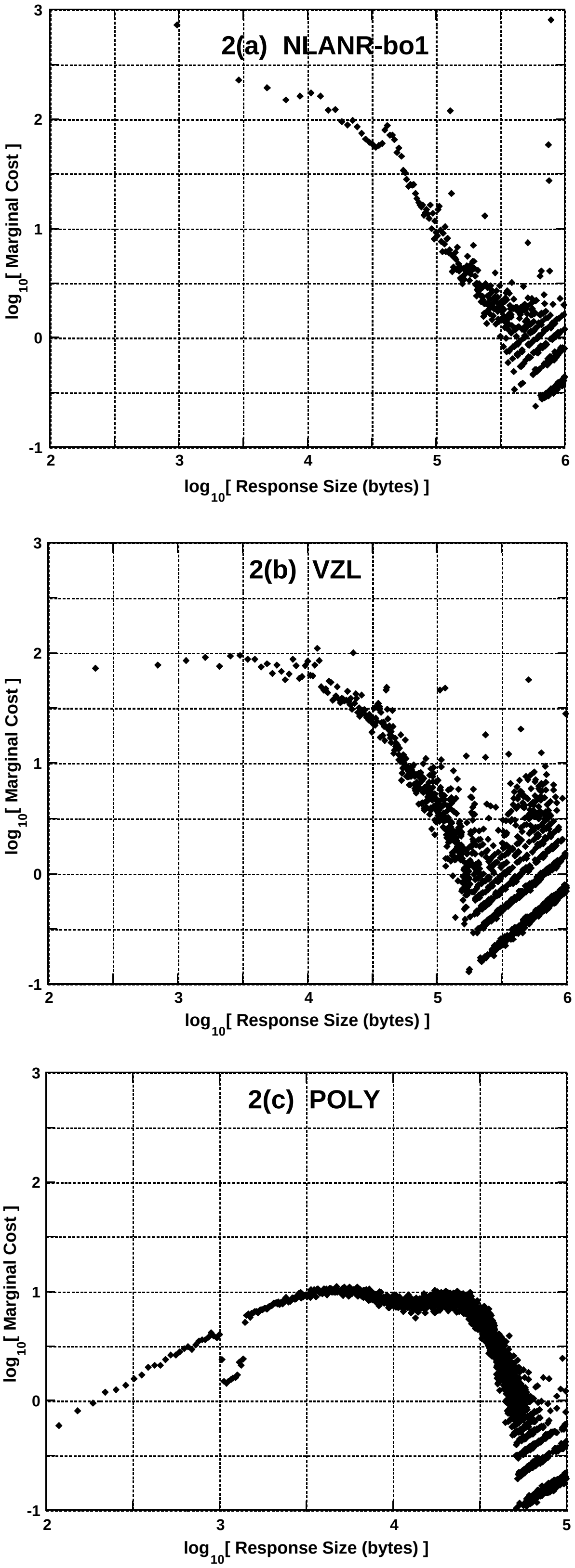

Figure 2: Estimated processing or bandwidth marginal cost as a function of file size.

marginal cost that is significant to larger file sizes, it gives more weight to large files than the NLANR-bo1 and VZL workloads. Applying these results to a trade-off analysis between storage and bandwidth costs requires quantitative estimates of their respective costs. Additional work on cost analysis is underway.

C. Null Hypothesis H_0 *Test*

The Bootstrap provides a unique method for performing the null hypothesis test for the two-sample problem. Suppose that we examine two independent random samples $\mathbf{z}=(z_1, z_2,\ldots, z_n)$ and $\mathbf{y}=(y_1, y_2,\ldots, y_m)$ drawn from two possibly different CDFs, *F* and *G*. Having observed ***z*** and ***y***, we want to test the null hypothesis H_0 that there is no difference between *F* and *G*, H_0 : F=G. The equality F=G means that *F* and *G* assign equal probabilities to all sets, $\text{Prob}_F\{A\}= \text{Prob}_G\{A\}$, where *A* is any subset of the common sample space of ***z*** and ***y***. If H_0 is true, then there is no difference between the probabilistic behavior of a random ***z*** or a random ***y***. This enables us to determine whether two datasets are statistically equivalent. It can also be used to evaluate the accuracy with which a fitted function represents an underlying distribution by applying the test to the original dataset and one calculated using that function.

We used the Bootstrap to test the null hypothesis H_0: F=G, where *F* is the NLANR-bo1 response size distribution with a random sample ***z*** of size *n* and *G* is the VZL or POLY response size distribution with a random sample ***y*** of size *m*. Let ***x*** be a combined sample of ***y*** and ***z***. We seek an achieved significance level:

$$\text{ASL} = \text{Prob}\ \{t(\mathbf{x}^*) \geq t(\mathbf{x})\}.$$

The quantity $t(\boldsymbol{x})$ is fixed at its observed value and the random variable $\boldsymbol{x}^*$ has a distribution specified by the null hypothesis H_0. Bootstrap hypothesis testing uses a substitution estimate for the distribution. The EDF of ***x*** puts equal probability on each member of ***x***. Under H_0, the EDF provides a nonparametric estimate of the common population that gave rise to both ***y*** and ***z***.

Appendix B describes an algorithm for computing ASL. In this paper, $t(\boldsymbol{x})$ is the difference between two-sample means and is chosen to be the Student's two-sample *t*-statistic, as described in Appendix B. The theoretical value is expected to be 50% for identical distributions. In row (8) of Table 1, the ASL for NLANR-bo1 is 48% because the NLANR-bo1 workload is the reference. Row (8) shows that NLANR-bo1 and the VZL possess the same response size distribution with a significance level of 15%. In contrast, the significance level for NLANR-bo1 and POLY having the same response size distribution is <0.01%, indicating that they are unlikely to be the same.

D. Distributions of File Popularity

We also investigated the variation of file popularity with popularity rank. File popularity is the relative frequency with which specific files are requested. This property has been extensively discussed in the literature, with some disagreement over whether the distribution is Zipf-like or not. A detailed examination of these issues can be found in Breslau *et al.* [5] and references therein. Since there has been a report of

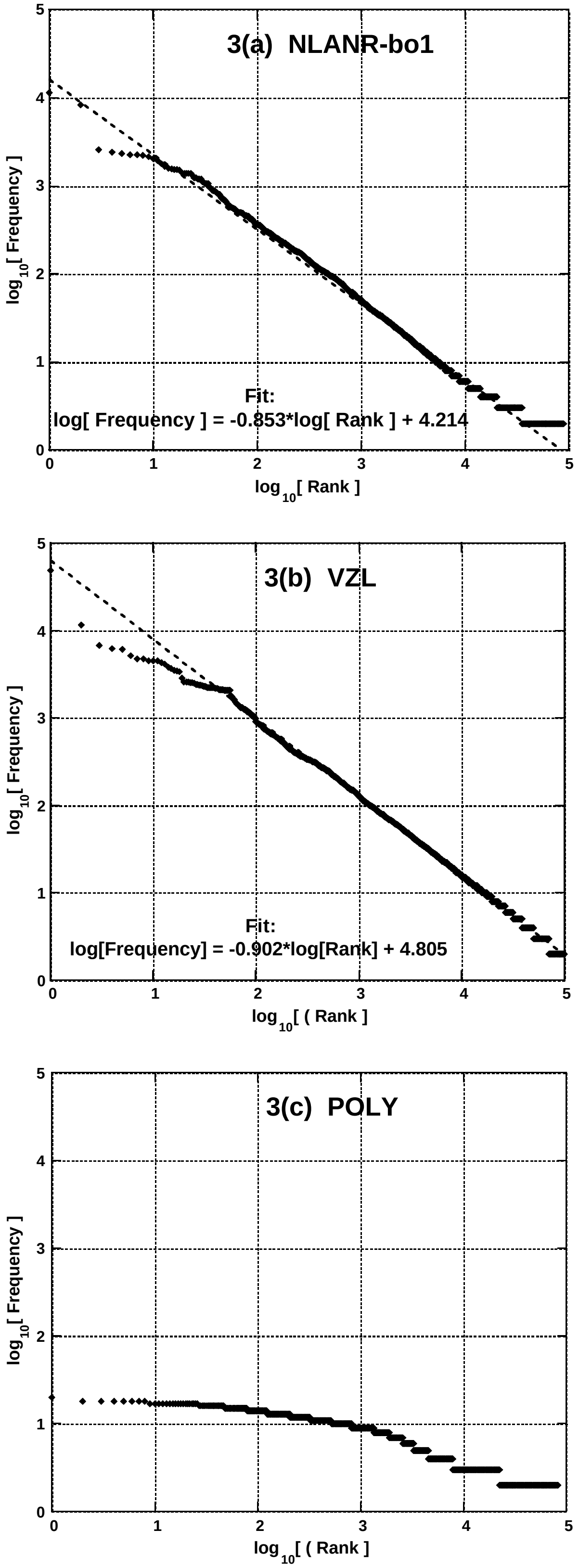

Figure 3: Frequency for requesting a file (popularity) as a function of file popularity rank.

popularity distributions changing over a period of years [3], our primary objective in this analysis was to compare our more recent data to results reported previously. Using the three log files, the number of requests (frequency) for each specific file was tabulated. Then the files were ordered from the most popular to the least popular and assigned a rank *r*. The rank is defined such that file *r* is the r^{th} most popular file with r = 1 the most popular file. Figures 3(a), (b), and (c) show for NLANR-bo1, VZL and POLY, respectively, the frequency of requests as a function of file popularity rank.

In Figures 3(a) and (b) for the real-world workloads, we see that for ranks larger than a relatively small number the frequency decreases linearly in the log-log plots, revealing inverse power law behavior. The exponents found from the fits shown are –0.853 for NLANR-bo1 and –0.902 for VZL. These values indicate the expected Zipf-like behavior and are similar to those reported previously [3,5]. In Figures 3(a) and (b) the frequency is observed to fall off much slower than Zipf's law for the 10-50 most popular files, which is also consistent with prior work [3,5]. The POLY data in Figure 3(c) does not exhibit Zipf-like behavior over any range for the Polymix-3 workload used. According to the Polygraph team, this is done intentionally to provide the required stress to the caches under test [19]. The roll-off in frequency for the least popular files is probably caused by the fixed working set size used by Polymix-3. This causes files to eventual age out of the set that can be requested. From this analysis we conclude that the NLANR-bo1 and VZL datasets have Zipf-like exponents typical of those observed over the past 5 years.

5. SUMMARY

We have used the Bootstrap methodology to analyze response size and file popularity distributions. The chief value of the Bootstrap is that it provides error estimates for statistics evaluated on nonparametric distributions, ones that are not well represented by mathematical expressions. In addition, it can be used to evaluate the statistical significance of how well a parameterized function represents the data. The data were drawn from three different proxies to obtain a measure of how workloads depend upon the client population. Included with the real-world workloads (NLANR-bo1 and VZL) was a synthetic workload generated by Web Polygraph (POLY), a widely used cache benchmarking tool, running Polymix-3.

We found the requested response size distributions for NLANR-bo1 and VZL to have pronounced heavy tails, consistent with previous reports. The NLANR-bo1 dataset had the heaviest tail while the synthetic POLY workload had the lightest tail. An analysis of marginal processing and bandwidth cost found it to fall off rapidly as a function of file size for NLANR-bo1 and VZL, but not for POLY. The Bootstrap null hypothesis test was employed to infer the significance level for the conclusion that the NLANR-bo1 distribution is the same as for the other two populations. We determined that NLANR-bo1 and VZL possess the same response size distributions with a significance level of 15%, where 50% is the maximum value. However, NLANR-bo1 and POLY have <0.01% significance level for having the same distribution. File popularity as a function of popularity rank has strong Zipf-like behavior for the NLANR-bo1 and VZL workloads for popularity ranks are larger than 10-50. By design, POLY does not exhibit Zipf-like behavior for the Polymix-3 workload.

From the analysis of proxy log files presented here, we conclude that the Bootstrap is a valuable tool for estimating errors in the values of important characteristics of workloads derived from actual traffic. Moreover, it can be used to evaluate the accuracy with which a synthetic workload represents actual Web traffic.

6. ACKNOWLEDGEMENTS

The authors are indebted to S. Zdonik for valuable discussion.

7. AUTHOR CONTACT INFORMATION

Johnson Lee	johnson.lee@verizon.com
William Miniscalco	bill.miniscalco@verizon.com
Meng Li	meng.li@verizon.com
W. David Shambroom	david.shambroom@verizon.com
John Buford	john.buford@verizon.com

APPENDIX A

THE BOOTSRAP METHOD FOR ESTIMATING STANDARD ERRORS

Suppose we encounter a common data analysis problem: a random sample

$$\mathbf{x} = (x_1, x_2, \ldots, x_n) \qquad \text{(A1)}$$

from an unknown probability distribution *F* has been collected and we wish to estimate a parameter of interest $\theta = t(F)$ [see Eq.(6)] on the basis of ***x*.**

The Bootstrap methods depend on the notion of the Bootstrap sample which is defined to be a random sample of size *n* drawn from $\hat{F}$ [see Eq.(4) and Eq.(5)]

$$\hat{F} \rightarrow \mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_n^*) \quad . \qquad \text{(A2)}$$

The star notation indicates that $\mathbf{x}^*$ is not the actual data set $\mathbf{x}$, but rather a randomized, or resampled, version of ***x***. Corresponding to a Bootstrap data set $\boldsymbol{x}^*$ is a Bootstrap replication of $\hat{\theta} = s(\mathbf{x})$, *i.e.*,

$$\hat{\theta}^* = s(\mathbf{x}^*) \quad . \qquad \text{(A3)}$$

Note that *s(x)* may be the plug-in estimate $t(\hat{F})$, but does not have to be. It can be defined as a complicated function if users wish to study the Bootstrap estimate bias.

The Bootstrap methods can be used to estimate the accuracy of $\hat{\theta} = s(\mathbf{x})$ by evaluating the standard errors. The algorithm is as follows:

1. Select *B* independent Bootstrap samples $\boldsymbol{x}^{*1}, \boldsymbol{x}^{*1}, \ldots, \boldsymbol{x}^{*B}$, each consisting of *n* data values drawn with replacement from ***x*** as in Eq. (A2). For estimating standard error, the number *B* is ordinarily in the range 25 – 200.
2. Evaluate the Bootstrap replication corresponding to each Bootstrap sample,

$$\hat{\theta}^*(b) = s(\mathbf{x}^{*b}) \ , \quad b=1,2,\ldots,B \ . \qquad \text{(A4)}$$

3. Estimate the standard error $se_F(\hat{\theta})$ by the sample standard deviation of the *B* replications

$$\hat{se}_B = \{ \sum_{b=1}^{B} [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B-1)\}^{1/2} \qquad \text{(A5)}$$

where $\hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b) / B$.

The limit of $\hat{se}_B$ as *B* goes to ∞ is the ideal Bootstrap estimate of $se_F(\hat{\theta})$. For more details please consult references [7,11].

APPENDIX B

THE BOOTSTRAP STATISTIC FOR TESTING F = G

1. Draw *B* samples of size *n*+*m* with replacement from ***x***. Call the first *n* observations ***z**** and the remaining *m* observations ***y****.
2. Evaluate *t(x)* on each sample,
   $t(\mathbf{x}^{*b}) = \bar{z}^* - \bar{y}^*$, b=1, 2,…, B .
3. Approximate ASL by $\#\{t(\mathbf{x}^{*b}) \geq t_{obs}\} / B$
   where $t_{obs} = t(\mathbf{x})$ the observed value of the statistic.

More accurate testing can be obtained by using a Student statistic. In the above test, instead of $t(\mathbf{x}) = \bar{z} - \bar{y}$, we could use

$$t(\mathbf{x}) = \frac{\bar{z} - \bar{y}}{\bar{s}\sqrt{1/n + 1/m}} \quad ,$$

where $\bar{s} = \{[\sum_{i=1}^{n}(z_i - \bar{z})^2 + \sum_{j=1}^{m}(y_j - \bar{y})^2] / [n+m-2]\}^{1/2}$.

This is the two-sample *t*-statistic. Please consult references [7,11] for more information.

REFERENCES

[1] R. J. Adler, R. E. Feldman and M. S. Taqqu, eds. *A Pratical Guide to Heavy Tails*, Statistical Techniques and Application, Birkhauser, Boston Basel Berlin, 1998.

[2] Paul Barford and Mark Crovella, Generating Representative Web Workloads for Network and Server Performance Evaluation, Proc. of 1998 ACM SIGMETRICS Intl. Conf., 1998.

[3] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, Changes in Web Client Access Patterns, World Wide Web **2**, pp. 15-28, 1999.

[4] S. Belczyk, private Communications.

[5] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, Scott Shenker, Web Caching and Zipf-like Distributions: Evidence and Implications, IEEE InfoCom, 1999.

[6] M. Crovella, C. Lindemann and M. Reiser, Internet performance modeling: the state of the art at the turn of the century, Performance Evaluation **42**, 91-108, 2000.

[7] A. C. Davison and D. A. Hinkley, *Bootstrap Methods and their Application,* Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1997.

[8] J. Dilley, M. Arlitt, Improving Proxy Cache Performance:Analysis of three Replacement Policies, IEEE Internet Computing, Nov/Dec 1999.

[9] B. Efron, Bootstrap methods: another look at the jackknike, Ann. Statist. **7**, 1-26, 1979.

[10] B. Efron, Computers and the theory of statistics: thinking the unthinkable, SIAM Review **21**, 460, 1979.

[11] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability **57**, Chapman & Hall/CRC, 1993.

[12] J. Kangasharju, K. W. Ross and J. W. Roberts, Performance Evaluation of Redirection Schemes in Content Distribution Networks, Computer Comm. **24**, 207-214, 2001.

[13] Kihong Park, Gi Tae Kim and Mark E. Crovella, On the Effect of Traffic Self-Similarity on Network Performance, Proceedings of the SPIE International Conference on Performance and Control of Network Systems, Nov. 1997.

[14] A. Mahanti, C. Williamson and D. Eager, Traffic Analysis of a Web Proxy Caching Hierarch, IEEE Network **14**, 16-23, 2000.

[15] National Lab of Applied Network Research (NLANR). Sanitized access log. ftp://ircache.nlanr.net/Traces/ .

[16] S. Paul and Z. Fei, Distributed Caching with Centralized Control, Computer Comm. **24**, 256-268, 2001.

[17] Visual Numerics, *MSIMSL Math/Library*, Visual Numerics, Houston, Texas, 1994.

[18] Web Polygraph, http://polygraph.ircache.net/ .

[19] Web Polygraph, http://www.web-polygraph.org/reference/models/realism.html .