Analyzing the Long Range Dependence and Object Popularity in Evaluating the Performance of Web Caching

G. P. Sajeev, Government Engineering College, Kozhikode, India
M. P. Sebastian, National Institute of Technology, Calicut, India

ABSTRACT

Web cache systems enhance Web services by reducing the client side latency. To deploy an effective Web cache, analysis of the traffic characteristics is indispensable. Various reported results of traffic analysis show evidences of long range dependence (LRD) in the data stream and rank distribution of the documents in Web traffic. This article analyzes Web cache traffic properties like LRD and rank distribution based on the traces collected from NLANR (National Laboratory of Applied Network Research) cache servers. Traces are processed to investigate the performance of Web cache servers and traffic patterns. Statistical tools are utilized to measure the strengths of the LRD and popularity. The Hurst parameter, which is a measure of the LRD, is estimated using various statistical methods. It is observed that the presence of LRD in the traffic is feeble and does not influence the Web cache performance.

Keywords: Long Range Dependence, Measurement, Performance Evaluation, Rank Distribution, Traffic, Web Caching

INTRODUCTION

Web caching is emerged as a technique to leverage the quality of Web services (Barish & Obrazcza, 2000). In spite of the evolution of new caching schemes and algorithms, the performance of cache services never reached to the expected levels, as can be seen from Table 1, 2 and 3. Suitable setting of the cache performance parameters can provide a solution to this problem. Hit ratio, byte-hit ratio, client side latency and network load reduction (server side and network) are some important cache performance parameters. These parameters can be estimated by analyzing the input traffic stream and Web proxy traces.

A simple Web cache caches all cache-able objects, which are referenced through it. This decision is not influenced by the traffic pattern or by the physical location of the cache. Thus, the traditional caching algorithms normally fill cache with irrelevant pages. Cache itself may
Table 1. Traces used

<table>
<thead>
<tr>
<th>Trace</th>
<th>Date</th>
<th>Number of requests</th>
<th>Hit Ratio</th>
<th>Zipf slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLANR-pa</td>
<td>1st Sep 2007</td>
<td>2,80,062</td>
<td>36%</td>
<td>0.72</td>
</tr>
<tr>
<td>NLANR-pa</td>
<td>1st Oct 2007</td>
<td>2,10,734</td>
<td>32%</td>
<td>0.76</td>
</tr>
<tr>
<td>NLANR-bo2</td>
<td>10th Aug 2007</td>
<td>2,43,356</td>
<td>26%</td>
<td>0.69</td>
</tr>
<tr>
<td>NLANR-sj</td>
<td>9th Jan 2007</td>
<td>5,44,356</td>
<td>20%</td>
<td>0.80</td>
</tr>
<tr>
<td>Internet Traffic Archive</td>
<td>23rd Jan 2005</td>
<td>28,338</td>
<td>48%</td>
<td>0.72</td>
</tr>
</tbody>
</table>

Table 2. Summary statistics of ircache.net on May 2007

<table>
<thead>
<tr>
<th>Servers</th>
<th>sj.nlanr.net</th>
<th>uc.nlanr.net</th>
<th>bo2.nlanr.net</th>
<th>sd.nlanr.net</th>
<th>pa.nlanr.net</th>
<th>rtp.nlanr.net</th>
<th>ny.nlanr.net</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Requests</td>
<td>252276</td>
<td>238156</td>
<td>114426</td>
<td>1093569</td>
<td>136211</td>
<td>1475734</td>
<td>269663</td>
</tr>
<tr>
<td>Hit rate (docs)</td>
<td>42%</td>
<td>31%</td>
<td>35%</td>
<td>8%</td>
<td>18%</td>
<td>18%</td>
<td>18%</td>
</tr>
<tr>
<td>MEAN Obj Size</td>
<td>10255</td>
<td>15722</td>
<td>18636</td>
<td>42719</td>
<td>11603</td>
<td>32801</td>
<td>12569</td>
</tr>
<tr>
<td>MB served (all)</td>
<td>2467</td>
<td>3571</td>
<td>2033</td>
<td>44552</td>
<td>1507</td>
<td>46164</td>
<td>3232</td>
</tr>
<tr>
<td>MB served cache</td>
<td>215</td>
<td>243</td>
<td>166</td>
<td>2651</td>
<td>67</td>
<td>2850</td>
<td>263</td>
</tr>
<tr>
<td>Percent Savings</td>
<td>9%</td>
<td>7%</td>
<td>8%</td>
<td>6%</td>
<td>5%</td>
<td>6%</td>
<td>8%</td>
</tr>
</tbody>
</table>

Table 3. Size Hit ratio performance sj ircache.net January 2007

<table>
<thead>
<tr>
<th>Size</th>
<th>Total (requests)</th>
<th>Misses</th>
<th>Hit%</th>
<th>Total (volume)</th>
<th>Misses</th>
<th>Hit%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-0.1KB</td>
<td>61</td>
<td>61</td>
<td>0.00%</td>
<td>318B</td>
<td>318B</td>
<td>0%</td>
</tr>
<tr>
<td>0.1-1.0KB</td>
<td>232.353K</td>
<td>58.344K</td>
<td>32.00%</td>
<td>97.356 MB</td>
<td>70.411MB</td>
<td>8.00%</td>
</tr>
<tr>
<td>1-5 KB</td>
<td>195.171K</td>
<td>174.626K</td>
<td>11.00%</td>
<td>443.825 MB</td>
<td>393.417 MB</td>
<td>11%</td>
</tr>
<tr>
<td>5-10 KB</td>
<td>41.556K</td>
<td>35.611K</td>
<td>14.00%</td>
<td>284.013 MB</td>
<td>244.657 MB</td>
<td>14%</td>
</tr>
<tr>
<td>10-50KB</td>
<td>54.226K</td>
<td>47.719K</td>
<td>12.00%</td>
<td>1.147GB</td>
<td>1.010 GB</td>
<td>12%</td>
</tr>
<tr>
<td>50-100KB</td>
<td>6.794K</td>
<td>6.289K</td>
<td>7.00%</td>
<td>456.296 MB</td>
<td>423.437 MB</td>
<td>7%</td>
</tr>
<tr>
<td>100-500KB</td>
<td>8.531K</td>
<td>5.209K</td>
<td>39.00%</td>
<td>2.376 GB</td>
<td>1.010 GB</td>
<td>58.00%</td>
</tr>
<tr>
<td>0.5-1.0MB</td>
<td>1.562K</td>
<td>1.542K</td>
<td>1.00%</td>
<td>1.090 GB</td>
<td>1.075 GB</td>
<td>1.00%</td>
</tr>
<tr>
<td>1-5 MB</td>
<td>3.231K</td>
<td>3.198K</td>
<td>1.00%</td>
<td>6.970 GB</td>
<td>6.904 GB</td>
<td>1.00%</td>
</tr>
<tr>
<td>5-10 MB</td>
<td>120</td>
<td>110 MB</td>
<td>8.00%</td>
<td>818.308 MB</td>
<td>738.650 MB</td>
<td>10.00%</td>
</tr>
<tr>
<td>10-50 MB</td>
<td>71</td>
<td>68 MB</td>
<td>4.00%</td>
<td>1.219 GB</td>
<td>1.176 GB</td>
<td>4%</td>
</tr>
<tr>
<td>50-100MB</td>
<td>17</td>
<td>17</td>
<td>0.00%</td>
<td>1.339 GB</td>
<td>1.339 GB</td>
<td>0.00%</td>
</tr>
<tr>
<td>≥ 100MB</td>
<td>11</td>
<td>11</td>
<td>0.00%</td>
<td>1.810 GB</td>
<td>1.810 GB</td>
<td>0%</td>
</tr>
<tr>
<td>Grand Total</td>
<td>544.370K</td>
<td>432.805K</td>
<td>20.00%</td>
<td>18.003 GB</td>
<td>1.851 GB</td>
<td>10%</td>
</tr>
</tbody>
</table>
become a bottleneck for the end user when all his requests are routed through the cache. An effective cache system must terminate majority of the requests within it, thus reducing end-user latency and load on the network. To reap the full benefit of caching, a cache system must be able to cache and preserve the pages which are on demand (Jin & Bestavros, 2000). To fill the cache with relevant pages, the cache designer must study the request streams characteristics.

It has been reported (Breslau, Cao, Fan, Phillips, & Shenker, 1999; Krashakov, Teslyuk, & Shchur, 2006) that the nature of Web access follows Zipf-like power law (Zipf, 1929). The introduction of proxy-cache service between Web client and the Web server has not improved the situation. That is, though the Web traffic is routed through cache servers, the access nature of user community remains more or less unchanged. This finding helps to identify the objects which are on demand and popular for a user community.

Self-similar and long range dependence (LRD) properties in Web traffic came into the attention of researchers when it was discovered that Ethernet traffic exhibits these properties (Leland, Taqqu, Willinger, & Wilson, 1994). LRD is a significant property in optimizing buffer sizes, admission and congestion control in LAN traffic (Rezaul & Grout, 2007). Similar observations were made for Web traffic characteristics also (Crovella & Bestavros, 1996).

This article focuses on studying the Web traffic characteristics and its impact on the Web cache parameters. The previous related works were either to search for a new characteristic or to improve the system using a known characteristic (Khayari, 2006). Our objective is to compare the strengths of these characteristics with real and synthetic traffic traces. Also, we intend to determine the influence of these characteristics on the Web cache performance parameters. First we check the traffic streams for its self-similar nature. If the traffic is self-similar, we measure the strength of LRD in terms of the Hurst parameter. Then, the experiments are carried out with the dataset of real traffic, to evaluate the performance parameters such as Hit Ratio (HR) and Byte Hit Ratio. A second round of experiments is done with the same dataset, but with induced LRD. Then the cache performances are compared. Similarly, measurements are carried out for the rank distribution. Further, we determine the effect of the rank distribution on cache performance. Thus, the main objective of this research is to find the weaknesses in the general characterization of the Web cache traffic, and its impact on the cache performance.

The remainder of the article is organized as follows. In the next section, we present related works. A subsequent section narrates the basics of long range dependence and rank distribution. Then, the data analysis using the collected trace files is presented. Following the data analysis section, simulation setup and results are given. Finally, we conclude the article with some directions for future research.

RELATED WORKS

The properties like self-similarity, LRD and rank distribution were observed and reported in various domains and contexts by the researchers. G. K. Zipf (1929) observed rank distribution of words in English dictionary. The self-similarity was observed by Edwin Hurst in the flood levels of the Nile (Liu, Wang, & Qu, 2005). The rank distribution of Web objects has been reported in many publications. Breslau (Breslau et al., 1999) is carried out an extensive trace analysis and observed that Web access nature follows a Zipf-like law. Dolgikh and Sukhov (2002) have done theoretical study of cache system based up on Zipf-like law. Krashkov et al. (2006) observed that Web site popularity can be explained using Zipf-like law. Crovella and Bestavros (1996) found that self similarity in HTTP traffic can be explained using Web document’s size distribution. Dill, Kumar, McCurley, Rajagopalan, Sivakumar, and Tomkins (2002) discovered Web’s fractal behavior in several senses, scales and folds. His work revealed the feasibility of modeling the Web as a graph. A
new classification algorithm is formulated in Khayari (2006) using the self-similarity and heavy-tail nature of the Web traffic. An extensive trace study has been done in Cesar and Deni (2007) to explore the self-similarity. The time domain analysis of Web traffic characteristics is carried out in Bai and Williamson (2004) and they suggested that the approximate model to capture the properties of Web traffic is gamma distribution.

LONG RANGE DEPENDENCE AND RANK DISTRIBUTION

This section reviews the fundamental principles of LRD and rank distribution. Kolmogorov first reported the self-similar process in 1941. Improvements were suggested by a number of researchers (Bai & Williamson, 2004; Cesar & Deni, 2007; Crovella & Bestavros, 1996; Gong, Liu, Misra, & Towsley, 2005; Pacheco & Roman, 2006).

Self-Similarity

A stochastic process is self-similar, if it is invariant in distribution of time and space. Consider a m-aggregated time series

\[ X^{(m)} = \sum_{n} X^{(m)}_{k} \quad k = 1, 2, 3, \ldots \]  

That is, \( X^{(m)} \) is obtained by summing the series \( (X^{(m)}_{k}, \quad k = 1, 2, 3, \ldots) \) over non overlapping block of size m. Hurst parameter is a measure of self-similarity and LRD. The time series \( X \) is self-similar, if for all positive m, the same distribution as \( X \) is rescaled by \( m^{\beta} \). That is,

\[ X_{i} = \frac{1}{m^{\beta}} \sum_{i=(i-1) m + 1}^{i m} X_{i}. \]  

The auto-correlation function (ACF) of \( X \) is

\[ r(k) = \frac{E[X_{i} - \mu][X_{i+k} - \mu]}{\sigma^2}. \]  

We say that, the series \( X \) is having self-similarity, if \( r(k) \) is same for \( X^{(m)} \quad \forall m \). The series with self-similarity may exhibit long-range dependence when,

- The auto-correlation function plot with respect to m will not decay rapidly.
- \( r(k) \) follows a power law \( \Rightarrow r(k) \sim k^{-\beta} \), where \( 0 < \beta < 1 \).
- The summation of auto-correlation function tends to infinity.

MEASURE OF SELF-SIMILARITY AND LRD

We consider seven standard methods to test self-similarity (Crovella & Bestavros, 1996; Leland et al., 1994). In that, five are graphical methods and two are of non-graphical type. We examine each of these methods in brief.

- **Absolute value method:** In this method an aggregated series \( X^{(m)} \) is defined, using different block sizes m. The log-log plot between the aggregation level and the absolute first moment of the aggregated series \( X^{(m)} \) is a straight line with slope of \( H - 1 \), if the data is long-range dependent.

- **Variance-time plot:** This plot is between variance of \( X^{(m)} \) and m on a log-log scale. The slope of the straight line is denoted as \( \beta \). If the \( \beta \) is greater than -1, then the data series possess LRD.

- **R/S plot:** The rescaled range of data series and R/S plot was developed by Edwin Hurst (Pacheco & Roman, 2006). He established that for many naturally occurring processes, the expected value of R/S, is proportional to \( r^{H} \), where n is the size of the data set and H is the Hurst parameter. If \( H \) value is greater than 0.5, the data series is having long-term dependent structure.

- **Periodogram method:** This method uses a frequency domain approach. The plot is between the power spectrum of the data set and frequency. The slope of the...
straight line of the Periodogram plot is \( \beta - l = 1 - 2H \), close to the origin.

- **Whittle estimator**: This method also belongs to the frequency domain, and is based on the minimization of a likelihood function, which is applied to the Periodogram of the time series. It gives an estimation of \( H \) and produces the confidence interval. It does not produce a graphical output.

- **Variance of residuals**: A log-log plot of the aggregation level and the average of the variance of the residuals of the series is a straight line with slope of \( 2H \).

- **Abrây-Veitch**: Wavelets are used to estimate the Hurst exponent. The energy of the series in various scales is studied to provide an estimate.

Most of the earlier researchers (Cesar & Deni, 2007; Gong et al., 2005) used the terms self-similarity and LRD interchangeably. In Pacheco and Roman (2006), clear distinctions between these terms are explored. We assume similar meaning for both the terms and define a new terminology **strong-LRD** for describing the traffic nature.

- **Strong-LRD**: A data series possesses strong-LRD if the Hurst parameter \( H \) takes a value close to 1. Precisely, \( 0.75 < H < 1 \). Our definition is justified according to Gong et al. (2005). When \( H \) tends to 1, the series becomes perfect self-similar.

**Document Rank Distribution**

The document distribution for request approximately follows a form of Zipf's law (Breslau et al., 1999; Krashakov et al., 2006; Serpanos, Karakostas, & Wolf, 2000) and is given by

\[
P_k(\alpha, N) = \frac{1}{H_N^{\alpha}}
\]

with

\[
H_N = \sum_{i=1}^{N} \frac{1}{k^\alpha}
\]

where \( P_k \) is the probability that \( k \)th ranked document will be accessed from \( N \) documents under consideration, and \( \alpha \) is some parameter between 0 and 1. We assume that \( k \) is defined for \( i=1, 2, \ldots N \). In other words, Zipf's law allows one to calculate the number of accesses to each object based on its popularity. Zip function is used to quantify the probability that access is made to the object \( O_k \), where \( k \) is the rank of the object.

Let us have a closer examination of the Zipf law. We follow (Breslau et al., 1999; Krashkov et al., 2006; Serpanos et al., 2000) and use a simple form of Zipf law

\[
P_i = \frac{A}{i^\alpha}
\]

where \( P_i \) is the probability of accessing \( i \)th popular object and \( A \) is a constant related to the number accesses to the highest ranked object. Intuitively, \( A = P_{fist} \) the probability that most popular object will be requested. Since the sum of the probabilities is equal to 1,

\[
\sum_{i=1}^{N} P_i = 1 \Rightarrow A \times H_N = 1
\]

Thus

\[
A = \frac{1}{H_N} \approx \frac{1}{\ln N}
\]

Where \( H_N \) is the \( N \)th harmonic number which is approximated to \( \ln N \). Now

\[
P_i = \frac{A}{i^\alpha} = \frac{1}{H_N \times i^\alpha} \approx \frac{\ln N}{i^\alpha}
\]

If the number of accesses \( N_a \) is directed towards the set of \( N \) objects, then the object \( O_i \) will be accessed \( P_i \times N_a \) times. Then the number accesses towards \( k \) popular objects is given by

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
\[ \sum_{i=1}^{k} = N_A \times \frac{H_k}{H_N} \]  

Then the document hit ratio is computed as

\[ h_d = \frac{N_A \times H_k}{H_N} = \frac{H_k}{H_N} \]  

Also,

\[ H_k = H_N \times h_d \approx \log k. \]

Hence,

\[ k = e^{h_d \times H_N} \]  

Thus, one computes, \( k \), the number of popular objects required in the cache to achieve a hit ratio of out of \( N \) objects. In other words, the cache size can be determined for a given hit ratio.

**SITE POPULARITY**

According to Krashakov (2006), the Web site accesses can also be modeled in the form of Zipf-like law with a modified rule (Zipf-Mandelbrot law). The problem with pure Zipf law is trickle-down effect (Doyle, Chase, Gadde, & Vahdat, 2002)—it will not hold true in tails or small rank region. The authors (Krashakov et al., 2006) proposed an offset parameter to correct this problem.

Now we extend this result to analyze the cache parameters with respect to the Web sites. Let there be \( M \) requests toward Web sites. The probability that a Web site with rank \( \omega \) will be accessed can be written in the form of Zipf law as

\[ R_{\omega} = \frac{\Omega}{\omega^\alpha} \]  

where \( R_{\omega} \) is the number accesses towards \( \omega \)th popular site and \( \Omega \) is a parameter similar to \( A \) in equation (5). Extending equations (6) through (10), the Web site hit ratio is formulated as

\[ h_w = \frac{H_w}{H_M}. \]  

Also,

\[ w = e^{h_d \times H_M}. \]  

That is, for a given hit ratio \( h_w \), one can compute \( \omega \) popular Web sites pertaining to a user community. The parameter \( \omega \) can be used for Web site mirroring and for maintaining DNS records (Krashakov et al., 2006).

Note that a Web site hit on the cache server need not be a document hit. It only implies some documents of the corresponding Web site are stored in the cache. Without losing generality, we assume that the average number of documents in \( w \) popular Web sites is \( \mu \). Then the total number of documents pertaining to \( \omega \) Web site is \( \omega \times \mu \).

By combining equations (10) and (13), the aggregate hit ratio is computed as

\[ h = \frac{H_k}{H_N} + \frac{H_w}{H_M} - \left( \frac{H_k}{H_N} \times \frac{H_w}{H_M} \right). \]  

Application of Zipf-like law yields a quantitative approximation regarding number of accesses towards Web objects and Web sites.

**Data Collection and Analysis**

We have collected traces, sanitized logs and summary statistics from NLANR caches (NLANR, 2007) during the period January 2005 to May 2008 and analyzed the traces for access behavior of users and Web cache performance. This collection enabled us to generate an enough workload for the simulation runs. To generate the synthetic work load, we used the same dataset. With help of software tools (Karagiannis, Faloutsos, & Mbole, 2003; Markatchev & Williamson, 2004), LRD is induced in the dataset. This is for generating a
workload which is independent of any process. Note that most of the simulation tools of cache systems generally use Autoregressive Integrated Moving Average (ARIMA) class of process to generate workload.

We present some of the traces and their characteristics. Table 1 depicts the trace, period and their properties. Table 2 is the performance summary of IR cache (NLANR, 2007) on a particular day, and Table 3 gives a detailed analysis of the trace. To know about the quantitative performance of Zipf-law (Krashkov et al., 2006) for Web sites, we have enlisted the rank of a particular Web site for a period of one year, in Table 5.

**OBSERVATIONS ON SELF-SIMILARITY AND LRD**

We have processed the data from the collected trace files, in number of ways by making use of various tools (Karagiannis et al., 2003; Markatchev & Williamson, 2004; R Development Core Team, 2008). Initially, we used the R language package (R Development Core Team, 2008) to know about the raw data behavior. We have plotted ACF and R/S plots in Figure 1, and found that the behavior is not perfect self-similar, but *only self-similar*. Also the R/S diagram reveal evidences of self-similarity. To measure the Hurst parameter, we used the Selfis tool (Karagiannis et al., 2003) and found that Hurst values are in the range of 0.312 to 0.572. For large trace file that we have analyzed (NLANR-sj) with 5,44,356 number of requests and the measured Hurst values ranges between 0.412 and 0.675.

Table 4 shows traces and the corresponding Hurst values. Figure 2 shows the Hurst value estimation of various methods. We did not observe strong-LRD in any of the data sets.

**Rank Distribution of Documents and Sites**

We have analyzed the collected traces for rank distribution of Web documents and Web sites. The observations by (Breslau et al., 1999; Krashkov et al., 2006; Meiss, Menczer, Fortunato, Flammini, & Vespignani, 2008) are true in the above case too with small variations. Here a very large number of requests are directed towards a single object. For example, in the trace we have analyzed, 6% of the requests are for a single object (in sj.ircache.net, 32563 requests out of 544356).

In Figure 3, the frequency spectrum of document access is plotted, by taking 100 documents as a group in rank-wise. The expected values of same zipf slope is also plotted. If we consider the quantitative measurement in this case, the difference is around 30,000. In Zipf law, the number of accesses to a particular docu-

<table>
<thead>
<tr>
<th>Estimation method</th>
<th>pa.nlshr.net(1)</th>
<th>pa.nlshr.net(2)</th>
<th>bo2.nlshr.net</th>
<th>sj.nlshr.net</th>
<th>Internet Traffic Archive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance</td>
<td>0.458</td>
<td>0.315</td>
<td>0.484</td>
<td>0.525</td>
<td>0.256</td>
</tr>
<tr>
<td>RS plot</td>
<td>0.458</td>
<td>0.498</td>
<td>0.0458</td>
<td>0.509</td>
<td>0.290</td>
</tr>
<tr>
<td>Absolute Moment</td>
<td>0.519</td>
<td>0.490</td>
<td>0.519</td>
<td>0.587</td>
<td>0.320</td>
</tr>
<tr>
<td>Variance of Residuals</td>
<td>0.312</td>
<td>0.134</td>
<td>0.302</td>
<td>0.412</td>
<td>0.297</td>
</tr>
<tr>
<td>Peridogram</td>
<td>0.564</td>
<td>0.474</td>
<td>0.564</td>
<td>0.625</td>
<td>0.319</td>
</tr>
<tr>
<td>Whittle</td>
<td>0.502</td>
<td>0.574</td>
<td>0.541</td>
<td>0.675</td>
<td>0.380</td>
</tr>
<tr>
<td>Abry- Veitch</td>
<td>0.572</td>
<td>0.500</td>
<td>0.522</td>
<td>0.636</td>
<td>0.382</td>
</tr>
</tbody>
</table>

Table 4. Hurst value estimation using various methods.
Table 5. Summary statistics from ircache.nlannr.net, of Website friendster.com for a period of one year

<table>
<thead>
<tr>
<th>Date</th>
<th># of requests</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>10/09/05</td>
<td>22966</td>
<td>3</td>
</tr>
<tr>
<td>10/10/05</td>
<td>13591</td>
<td>5</td>
</tr>
<tr>
<td>12/11/05</td>
<td>20565</td>
<td>4</td>
</tr>
<tr>
<td>10/12/05</td>
<td>31254</td>
<td>2</td>
</tr>
<tr>
<td>08/01/06</td>
<td>29656</td>
<td>3</td>
</tr>
<tr>
<td>12/02/06</td>
<td>28184</td>
<td>12</td>
</tr>
<tr>
<td>10/03/06</td>
<td>23729</td>
<td>11</td>
</tr>
<tr>
<td>22/04/06</td>
<td>18838</td>
<td>7</td>
</tr>
<tr>
<td>22/05/06</td>
<td>not ranked</td>
<td>-</td>
</tr>
<tr>
<td>01/06/06</td>
<td>17473</td>
<td>12</td>
</tr>
<tr>
<td>01/07/06</td>
<td>56435</td>
<td>1</td>
</tr>
<tr>
<td>01/08/06</td>
<td>40372</td>
<td>5</td>
</tr>
</tbody>
</table>

Figure 1. LRD check using ACF and R/S plot

PERFORMANCE EVALUATION

Our simulations use NLANR traces and synthetic workload generated using WebTraf (Markatchev & Williamson, 2004). The simulations are carried out in NS2 and WebTraf environments. Trace analysis tools like traceconv (TraceGraph, 2005) are also used. Simulations were under different values of Zipf exponent, and varying cache size. As reported in (Breslau et al., 1999; Doyle et al., 2002), a good approximation of Zipf law holds true for a larger sample size in the order of $10^5$. We have
Figure 2. Hurst value estimation using various methods

Figure 3. Document access nature of bo2.nlarnr.net

taken sufficient number of references (20000 to 400000) in all the cases. For the real work load we have used the data set from NLANR traces (NLANR, 2007).

We have preprocessed the NLANR traces as described in (Cao & Irani, 2002). The NLANR traces contain IMS (if-modified-since), and non cacheable objects like cgi-bin. The preprocessor removed all such requests before generating the work load. Experiments are carried out using the original data set and a generated one (where LRD is induced). We have applied the
cache replacement strategies such as Least Frequently Used (LFU), Least Recently Used (LRU), Greedy Dual Size (GDS), First In First Out (FIFO) and Random (RAND). Figure 5 shows the hit ratio performance using original bo2 trace and Figure 6 shows the hit ratio performance using bo2 trace with induced LRD. LRD in the data set does not influence the cache hit rate performance as can be seen from Table 6. A noticeable change occurs only in the case of GDS policy (6% down). This is because; the induction of LRD in the data set may neutralize the size based policy. Note that LRD in the dataset offers uniform bursts in the traffic.

The effect of LRD on cache growth is plotted in Figure 7. The traces used for the simulation are with equal length and varying Hurst value. It is observed that the cache is being filled at faster rate for a trace with a larger Hurst value. The effect of rank distribution on hit ratio is in Figure 8. A larger hit ratio is achieved using a synthetic work load, which follows perfect Zipf-law.

**CONCLUSION**

Traffic measurements and analysis are needed for assessing the effectiveness of Web cache

---

Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Figure 6. Hit ratio performance of shuffled NLANR-bo2 trace with \( H = 0.864 \)

![Performance charts for bo2 trace with LRD and policies]

Table 6. Performance comparison of different cache replacement policies on LRD

<table>
<thead>
<tr>
<th>Policy</th>
<th>Original Trace</th>
<th>LRD induced trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRU</td>
<td>30%</td>
<td>29%</td>
</tr>
<tr>
<td>LFU</td>
<td>38%</td>
<td>37%</td>
</tr>
<tr>
<td>GDS</td>
<td>37%</td>
<td>31%</td>
</tr>
<tr>
<td>FIFO</td>
<td>30%</td>
<td>31%</td>
</tr>
<tr>
<td>Random</td>
<td>34%</td>
<td>32%</td>
</tr>
</tbody>
</table>

Figure 7. Showing the effect of LRD on cache growth

![Effect of LRD on Cache growth chart]

servers. In this article we have analyzed the Web cache traffic to study about LRD and rank distribution. From the trace analysis, it is found that the Web traffic is only self-similar, and the presence of LRD is feeble. The LRD component in the traffic does not affect the cache performance directly, but influences the cache growth. Most of the earlier research
Figure 8. Showing the effect of rank distribution

works on Web traffic were for characterizing it with some generic rules. Our measurements have revealed that a considerable portion of the Web cache traffic is heterogeneous and follows no specific rules like the Zipf-law. Though a general characterization of the Web traffic yield simplified caching models, it may not give precise results.

Our results could be useful to optimize the cache framework. Simulation tools for Web cache systems generally use Autoregressive Integrated Moving Average (ARIMA) class of method to generate the synthetic workload. In the light of new evidences, we suggest Generalized Autoregressive Conditional Heteroskedasticity (GARCH) type of method, to represent the irregular components of the traffic. Building a cache framework considering this irregularity of the Web cache traffic is suggested as a topic for future research.

REFERENCES


Copyright © 2009, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.