Structural Characterization of Popular Web Documents

Abdolreza Abhari, Sivarama P. Dandamudi
School of Computer Science
Carleton University, Ottawa
Ontario K1S 5B6,Canada
{abdy, sivarama}@scs.carleton.ca

Shikharesh Majumdar
Department of Systems and Computer Engineering.
Carleton University, Ottawa
Ontario K1S 5B6, Canada
majumdar@sce.carleton.ca

Abstract

Characterization of Web documents is essential to study performance issues such as minimizing demands on the back-end servers and communication overheads. In addition, characterization of the Web is also important to devise synthetic workload generators for use in the investigation of effective resource management algorithms. Most characterization of Web documents are based on Web files without considering their inherent structure. To display a complete Web page a collection of files that include the files corresponding to the embedded objects in a page must be transferred. A Web object is defined to be this collection, i.e., a Web page and its related embedded files. Our goal in conducting this study is to collect data on the structure and size of Web objects that is particularly useful in improving Web server performance through techniques such as clustering of files, parallel I/O, and data caching on client sites. We report the results of an empirical study conducted on several popular (in top 100 sites) Web sites. We have chosen the popular Web sites for this investigation because they are more likely to be efficiently designed. In addition, popular Web servers also account for a significant portion of the network traffic. We also study the trace of a busy proxy access log to characterize Web objects for regular Web environments.

Keywords: *Web file characterization, Web objects, Embedded objects, Web server performance, Resource management.*

1. Introduction

Improving performance of Web servers (reducing their response times for example) is an active research area [5,11]. Using parallel hardware on a Web server is one solution for reducing server response times. For example IBM has used the SP2 parallel system in the design of Web Servers for the 1998 Olympic Winter games [5]. A cluster of workstations that can share cache information and cache data between the nodes was used by the Alexandria Digital Library (ADL) for implementing a Web server [11]. Some other researchers have tried to reduce network bottlenecks through file caching in proxies and by using load balancing on multi-node Web servers [11, 2].

Fundamental to the goal of improving Web performance is an understanding of the World Wide Web workloads and the characteristics of Web-based applications. There has been a significant research effort on the characterization of Web workloads (see [1,3] for example). Most of this research is concerned with the development of models for simulating the traffic generated by a Web environment [3]. In addition, most of the existing research is based on logs captured on scientific Web sites. Therefore, this characterization is limited by the files that happened to be accessed during the log capture time at the specific sites. Characteristics of Web documents in terms of their components and size have received little attention.

In this paper, we present a study to characterize a set of popular Web documents. This set is based on the weekly announcement of popular Web sites. The files that belong to the front pages of popular Web sites are examined during three different periods of a year. A proxy access log belonging to NLANR [14] is also analyzed to capture dynamic access to Web objects across the sites.

Contributions of the paper: One distinctive feature of this study is our consideration of the *Web object*, which is the collection of a Web page and all its related embedded objects. Most existing works do not discuss the structure and size of Web objects. As discussed in section 5 this information is useful in investigation of caching policies at proxy or client sites. We have found that most of Web objects are small in size and can therefore be cached effectively at the client sites. This paper reports on a study of *popula*r Web documents. Designers of popular Web documents have tried to reduce the page download time through efficient design of Web pages. The structure of popular Web pages can therefore provide guideline for designing Web pages that give rise to small download times.

In most of the previous work on Web file measurements, due to their focus on Web traffic characterization, a traffic-based measurement method was used. That is, the files that were actually transferred by clients were studied. We have used a source-based measurement approach to examine the number of embedded objects in a Web object. This approach involves examining each Web page source file and includes all of the embedded files in the page irrespective of whether it was present in the access log. In analyzing the access log we use source-based measurement to recognize embedded objects and their corresponding web files. This produces a potentially more accurate information about the number of embedded objects in comparison to the traffic-based approach.

In traffic-based measurements, time thresholds are used to recognize embedded objects. In this method, therefore, the results depend upon the selected threshold values. In addition, this method only looks at the number of objects transferred as opposed to the number of objects in the page. A difference between these two numbers, for example, can occur due to aborts by the clients (who may not wait for the complete page to be downloaded and decide to follow a link on the page) or existence of different levels of caching. Information on association between a Web page (e.g., HTML main page) and the embedded files is important when we want to consider file clustering to improve performance. Clustering of related files can potentially reduce disk access latencies as well as communication times.

The remainder of this paper is organized as follows. Section 2 presents an overview of the related work. Section 3 describes the data on popular Web documents collected in our study. An analysis of the results is presented in Section 4. Section 5 discusses the implications of the data characteristics on resource management. We conclude the paper in Section 6.

2. Overview of Web Documents and Related Work

Web servers typically provide both static and dynamic pages. Static documents are stored and retrieved from the file system. Dynamic documents are created on-the-fly by server programs that are invoked when requests are made. Dynamic pages are essential for situations in which contents of Web pages are constantly changing. In this study our focus is on static Web documents.

Web document structure: Barford and Crovella [3] defined a *Web object* to be a collection of all the files that must be transferred to display the complete page. A *Web page* maybe an HTML source file containing pointers to zero or more *embedded objects*. Embedded objects are typically files containing images that users receive as part of the Web page. When a user clicks on a Web object, the corresponding Web page and related embedded objects are sent to the client browser.

2.1 Related Work

In this section, we present a brief overview of the literature related to our work. As mentioned in Section 1, we classify measurement techniques into two types: source-based and traffic-based. In the source-based method, each Web page source file is examined to obtain the data. This method provides data that are more accurate for our purposes. The other method involves looking at the traffic and deriving the data based on the observed Web traffic.

2.1.1 Measurements on Web Documents

Most of the existing research has adopted the traffic-based measurement approach. A representative set of work is discussed.

Traffic-based measurements: Most of the works are based on server access logs. Server access logs capture the results of requests made to the server. Arlitt and Williamson [1] analyzed the server access logs of six sites: three from universities, two from scientific research organizations, and one commercial site. They measured the files that were successfully transferred and found that the distribution of file sizes is heavy tailed. This implies that the server's file system must deal with highly variable file sizes. A heavy tailed distribution is one whose upper tail declines following a power law. A distribution is heavy tailed if $P[X > x] \sim x^{-\alpha}$, as $x \to \infty$, $0 < \alpha < 2$. If $\alpha \leq 2$ then the distribution has infinite variance whereas $\alpha \leq 1$ indicates that the distribution has an infinite mean [6]. Simulation with a heavy tailed distribution has slow convergence to steady state, and high variability at steady state. In practice, the sample mean of random variables that follow a heavy tailed distribution can be achieved after many small observations mixed in with a few large observations [7].

Arlitt and Williamson also show that file size distributions match the Pareto distribution [1]. The Pareto distribution, which belongs to the class of heavy tailed distributions, is characterized by the probability mass function: $p(x) = \alpha k^{\alpha} x^{-\alpha-1}$, $\alpha, k > 0$, $x \geq k$.

Crovella and Bestavros [6] analyzed client-based trace logs obtained by modifying a Mosaic Web browser source code. They mention that one of the reasons for seeing self-similarity in Web traffic is the heavy tailed distribution of document sizes. They also observe that despite the presence of some large files such as multimedia files, Web file systems are currently more biased towards small files.

Barford and Crovella [3] used a threshold of one second for distinguishing embedded objects. With this assumption, two file transfers that are separated by a time period of one second or less are considered to correspond to two different embedded objects of a Web object. As mentioned earlier, users often interrupt Web page transfers. Also clients request for objects that are not in their cache. So this method may not provide an accurate information on the number of objects in a Web page. Source-based measurements avoid such problems.

Source-based measurement: Wills and Mikhailov [18] studied popular sites to gather statistics on the rate and nature of changes for all the embedded objects and their implications on caching. They found that despite the fact that HTML resources change frequently there is a significant amount of reuse of images. They suggest that cache replacement policies need to associate an image with its container resource. If an image is no longer used by any container resource then it should be garbage collected and removed from the cache. They suggest policies for caching resources from the same site should take into account the inherent structure of the page and to discard unneeded image files when the page structure changes. They characterize the number of embedded images per page but don't report their sizes and data on other types of embedded objects. The authors have also suggested the use of Web object structure in caching [18].

In the results reported by the Open text index group, Web page sources were used [4]. The goal of their research was to design an efficient search engine for finding specific keywords in Web pages. In their study they have examined 20,000 HTML Web pages and reported on the number of embedded objects that is discussed in Section 4.2.

2.2 Impact of Web Objects

Effect on the Web traffic: Embedded objects that belong to a Web object influence the traffic pattern. With more embedded objects in a Web document more files are transferred. The time interval between transferred embedded objects consists of processing time spent by the browser, in parsing of Web files and preparing to start a new TCP connection. Nielson and his colleagues [16] investigated the impact of embedded objects on performance of the HTTP 1.1 protocol by changing the size of the objects. They showed that changing images from GIF or JPEG format to Portable Network Graphics (PNG) improves network performance when HTTP 1.1 is deployed. PNG has several advantages over GIF/JPEG. It produces smaller files in comparison to GIF/JPEG and renders quickly on the screen like GIF while producing higher quality pictures.

Effects on Web page download time: Structure of a Web object can affect the download time of a Web page. Efficient design of a Web page can reduce the page download time. One way to improve the performance is to simply find which embedded objects in the whole Web site are more popular and put those objects in the front Web page. This method was used in the redesign of IBM's Web page for the 1998 Olympic Winter games [5]. Microsoft has redesigned its Web page in 1999 to improve the download time. The size and the number of embedded objects in the new version of their homepage are decreased [17].

3. Data Collection and Methodology

Since popular Web documents are expected to have a strong impact on performance, our research is concerned with the characterization of popular Web documents. We have studied a number of Web sites. The site 100hot.com contains various statistics about top 200,000 Web sites. This site is responsible for reporting the most popular sites for each week [13]. For this study 3 data sets were used.

The first data set was gathered in March 1999 and includes 55 front pages consisting of 1051 files. These were front pages of top ranked sites out of 100 popular sites. The second data set was collected from 100 popular sites in November 1999 and is made up of 86 front pages that consists of 1490 files. The third data set was gathered in February 2000 and consists of 92 front pages out of 100 popular sites made up of 1511 files. These data sets are about the static structure of Web documents and we call them data on popular documents. We also examine part of NLANR access log recorded on 25 January 2000 that we call log data. It includes 27,546 request entries. After cleaning and reducing the log based on considerations that were suggested by Krishnamurthy and Rexford in [12], log data ended up with 2369 unique Web pages consisting of 6101 files. Log data does not correspond to popular Web sites and is studied to compare the characteristics of poplar sites with that of regular Web sites.

A program was written for generating statistics on the size of pages and their embedded objects. This program parses a Web page and finds the size, type and number of its embedded objects. For log data first we ran this program on 2369 Web pages to find Web objects and their related embedded objects. Based on the embedded objects information we search in the access log to see how many of these embedded objects are requested. We generated two series of statistics: one is based on source-based information on 2369 Web pages and the other is based on requested files that we call request-based information. We ran the program 1 week after the access log date, ignoring the cases where the names of embedded objects were changed.

4. Data Analysis

Only data on three data sets and access log are presented in this section. The implications of these data are discussed in the following section. We refer to the previous data on Web documents and our data on access log as the “regular data set ” and our data on popular sites as the “popular data set”.

4.1 Web Object Sizes

By analyzing the Web pages we observe that some pages have multiple instances of the same embedded objects such as arrows. In this paper a repeated embedded object is considered only once in the measurement of Web object size. This is because a repeated object in a Web page is stored and transferred only once. The actual number of embedded files is computed after eliminating repeated embedded objects.

By adding the sizes of embedded files, the size of a Web object can be obtained. The data on Web object sizes including the cumulative frequency of sizes are presented in Figure 1 and Figure 2. We observe in Figure 1 that the sizes of more than 90% of Web objects are less than 100 KB. The maximum Web object size is observed to be 285252 bytes and the means of Web object sizes are in the range of 50,000 to 60,000 bytes (see Table 1).

For the log data the results of our source-based measurement show that the maximum Web object size is 5,993,115 bytes and the mean Web object sizes of 77,319 bytes. In request-based measurement of log data the maximum Web object size is 1,329,907 bytes and the mean Web object sizes of 24,046 bytes. The important observation is that in both of these measurements approximately 90% of Web objects are less than 100 KB in size. This agrees with our observation on popular Web objects (see Figure 2). The reason for the difference between source-based and request-based measurements for the Web object size is explained in section 4.2.

Since Web object sizes in popular Web documents are characterized by a large variance (see Table 1) and because the size of each Web object is the sum of the file sizes that have heavy tailed distributions (see section 4.3) the resulting distribution of Web object sizes is also heavy tailed [15].

In this study we have used Aest, the software that was developed by Crovella and Taqqu [8] to measure the tail weight (α). Aest uses the scaling method to compute α and gives the result that can be observed with Gnuplot. The result of running Aest on log data gives an α of 1.233 for a source-based measurement of Web object size and an α of 1.227 for a request-based measurement of Web object size.

Since $\alpha<2$ the Web object size distribution in the log data is also heavy tailed which agrees with our observation on popular Web object size distribution. Being heavy tailed suggests that although most Web objects are small we can have some extremely large Web objects.

4.2 Number of Embedded Objects

The number of embedded objects in a Web page is the number of unique files that must be transferred to display the page at the client site. The statistics on the number of embedded objects are presented in Table 1, Figure 3, and Figure 4. Figure 3 presents data that corresponds to the 100 popular sites. From this data we observe that 95% of Web pages have less than 40 embedded objects. Figure 4, which shows the log data, confirms this result for both request and source-based measurements. As shown in Figure 4, for the request-based measurement, 95% of requested Web pages contain 0 to 10 embedded objects. For source-based measurement, 70% of Web pages contain 0 to 10 embedded objects. We have found that, on average, 2 out of 8 embedded objects belonging to a Web object are requested in the log data. As a result we have two different lines for source-based and request-based measurements on the Web object size and number of embedded objects in Figures 2 and 4. In the study by Crovella and Barford [3], they used a threshold of 1 second in the recognition of embedded objects. Based on this threshold they estimated the number of embedded objects and suggested a Pareto distribution with parameters of $k=1$ and $\alpha=2.43$. Note that since $\alpha>2$ the distribution is not heavy tailed. In a similar work by Deng [9], a threshold of 60 seconds was used for the recognition of embedded objects but no distribution was suggested.

In another study by Open text index group [4], the number of image references in the source Web pages was counted to determine the number of embedded objects. For our purposes, the information obtained by counting the number of image files is inadequate. First, an image file may be repeated in a Web page several times. Thus, counting the number of image files is not an accurate method for determining the number of embedded files. It should be noted that only one copy of an embedded object is maintained independent of the number of times it is used in a Web page. Secondly, there can be other types of embedded objects, which are not included in their study. Finally, they do not present data on the mean number of embedded images in a Web page.

Our data shows that the variance of the number of embedded objects in a front page is small. Observing the shape of the distribution of number of objects and by using the Hill’s estimator (see [10] for details) we conclude that the distribution of the number of embedded objects in popular Web documents is not heavy tailed.

The result of running Aest on our log data gives an α of 2.1 for source-based measurement and an α of 2.22 for request-based measurement on the number of embedded objects in a Web object. Since α is higher than 2 the distribution of the number of embedded objects for regular Web documents is not heavy tailed.

4.3 Size of Web Files

In the study of the first data set of popular sites, March 1999 (1051 picture and HTML files), the mean file size is observed to be 3256 bytes. The plot of cumulative frequency for file sizes demonstrated that 90% of Web files have a size less than or equal to 10,000 Bytes (see Figure 5). It is interesting to note that the coefficient of variation of Web file size is large. The larger coefficient of variation for the embedded object size maybe a result of a mix of text and picture files used in constructing a popular Web page.

In our data, we observe the heavy tailed characteristic in the Web file size distribution. A high variance and the presence of a few large values in file sizes (see Figure 5 and Table 1) indicate the distribution is heavy tailed. However, computing α and examining the shape of the distribution are also important in the recognition of heavy tailed distribution [15]. Running Aest on the popular data sets March 1999 (1051 Web files), November 1999 (1490 Web files) and February 2000 (1514 Web files) produces an α of 1.596, 1.52 and 1.45, respectively. Since $\alpha<2$ the popular Web file size distribution is heavy tailed. For files that are larger than 10000 bytes, Crovella and Bestavros [6] report an α of 1.27 for all files and an α of 1.59 for text files. Arlitt and Williamson [1] observed that the size of files larger than 1024 bytes are Pareto distributed with $0.40<\alpha<0.63$.

4.4 Type of Web Files

Embedded objects are mostly picture files with a GIF or JPEG format. In the first data set of popular files from 1051 observed embedded objects, 71 HTML text files account for 33% of the total size of Web objects and 980 picture files account for 67% of the total size of Web objects. The HTML file size mean is 16054.46 bytes whereas the picture file size mean is 2329.14 bytes. A mean of 9123.33 bytes and 10053.16 bytes are reported in [1] for text and picture files, respectively. If we compare the sizes of popular Web documents with sizes reported for regular Web documents, we observe that on an average:

- Picture files of popular Web files are smaller than the picture files of regular Web files
- Text files of popular Web files are larger than the text files of regular Web files

We have also studied popular Web pages with audio and video files with MP2, MP3, MIDI and AGIF formats. These Web pages that were selected from 100hot.com sites contain links to audio and video files. We examined 92 audio and video files from popular Web files. Statistics on their sizes are presented in Table 2.

Our data show that about 93% of the files are picture files and the remaining 7% are text files. A key difference between the characteristics of popular Web files and regular Web files is that the picture files popular Web documents tends to be much smaller in size (2300 bytes versus 10,000 bytes). An implication of this is that IP fragmentation and reassembly overheads are expected to be much smaller for the picture files of popular Web pages. On an Ethernet LAN, the picture files of popular Web pages require at most two fragments. The corresponding value for the regular Web pages, on an average, is eight. In addition, the smaller picture file size reduces the demand for the TCP send and receive buffers.

5. Implications of Characteristics on Resource Management

A number of observations on the structure and sizes of Web objects and files were presented in the previous sections. In this section we summarize the implications of these observations on system management and performance. Most of our discussion focuses on text and picture files.

Web object size: Measurements made on three sets of popular Web objects consisting of HTML text and picture files show that 90% of the popular Web objects are less than 100,000 bytes in length (see Figure 1) whereas the maximum size of a Web object is observed to be 285,252 bytes (see Table 1). This indicates that in most cases entire Web objects can be cached in main memory at client workstations and can thus be re-accessed with low latency. The reasonable size of the popular Web objects imply that multiple popular Web objects can be cached for different clients without making large demands on a global cache (proxy) serving multiple clients at a site.

Web file size: Most Web files are observed to be small. 90% of Web files are less than or equal to 10,000 bytes (see Table 1 and Figure 5). This indicates that clustering all the Web files that belong to the same Web page on contiguous disk blocks can significantly reduce disk access time, which in turn reduces client response time. The distribution of Web file sizes is observed to be characterized by a high coefficient of variation (see Table 1). A large variability in the Web file size implies that a small proportion of the available files is large whereas most of the files are small in size (see Figure 5). On a single disk-based server a large file can monopolize the disk for a large amount of time when it is being transferred to the client site. By spreading its constituent Web files across multiple disks the load is more evenly balanced and the "disk hogging" that occurred in the single disk system can be effectively controlled.

Number of embedded objects: Each Web page was observed to contain 16 to 18 embedded Web files on an average (see Table 1). Each of the embedded files needs to be transferred to the client for displaying the contents of the entire Web page. Thus the constituent Web files of a single Web object can be stored on multiple disks at the server. These disks can be accessed in parallel when the Web object is requested. If the files are very small the object can be divided into clusters each of which consists of multiple Web files allocated on consecutive disk blocks for reducing the cluster access time. The clusters can then be distributed over multiple disks and accessed in parallel. Development of such parallel I/O can significantly enhance system performance.

Distributing Web files that belong to a Web object over multiple disks can effectively balance the disk load. The distribution of Web file sizes is observed to be characterized by a high coefficient of variation (see Table 1). A large variability in the Web file size implies that a small proportion of the available files is large whereas most of the files are small in size (see Figure 5). On a single disk-based server a large file can monopolize the disk for a large amount of time when it is being transferred to the client site. By spreading its constituent Web files across multiple disks the load is more evenly balanced and the "disk hogging" that occurred in the single disk system can be effectively controlled. Using multiple disks for the storage of a Web object can reduce the queuing delay: although a particular disk may be busy the access of a Web object may start with files stored on another disk that may not be accessed currently. A mean of 17 embedded objects per page indicates a good potential for parallel access of embedded files.

Audio/Video: Although the sizes of audio/video files exhibit less variability in comparison to those of the picture and text files (see Table 1 and Table 2) their mean size is orders of magnitude higher than that of the text and picture files. Since the transfer of such a file can involve a significant amount of time separate disks are required at the server for retrieving other text and picture files when an audio-video file retrieval is in progress. Multiple disks are also required to support multiple transfers of audio/video files at the same time. If a single disk is incapable of producing the desired latency, parallel I/O is to be used. More stringent synchronization mechanisms may be required, however, in comparison to text and picture files because for maintaining the clarity of audio and video a proper sequence of audio or video data blocks need to be transferred within a short period of time.

6. Conclusions

The difference between this study and previous work is that in this paper Web objects were characterized. In previous work Web files were studied without considering the inherent structure of the Web page that contains multiple Web files. We study the Web objects in popular Web documents and compare our results with the characterization of Web objects in regular Web documents. Since popular Web documents are accessed more often than others, their characteristics are expected to have a strong impact on performance. A summary of the important characteristics of Web object for popular and regular Web documents is presented in Table 3.

We found that the mean number of embedded objects for popular Web documents is approximately 17. Also, we showed that the distribution of the number of embedded objects is not heavy tailed, which agrees with the results reported by other researchers on regular Web sites. Little data exist on measurement of Web object size and characteristics of its distribution. In this study, the distribution of Web object sizes is observed to be heavy tailed and 90% of them have a size that is less than 100,000 bytes. Our data on the number of embedded objects and Web objects sizes can be useful in the investigation of Web object caching and prefetching techniques. The distribution of popular Web file sizes is heavy tailed and agrees with the previous observations on regular Web documents. We observe that text files in popular Web documents are larger in size than text files of regular Web documents reported in [1]. On the other hand, picture files of popular Web documents are smaller in size than picture files of regular Web documents described in [1].

We have also discussed the utility of the results reported in this paper. We specifically suggested how the characteristics reported could be exploited in resource management on Web servers by using file clustering and parallel I/O techniques. We are currently investigating these issues.

Acknowledgements

Financial support for this research was provided by Nortel Networks, Natural Science and Engineering Research Council (NSERC) of Canada and Carleton University.

References

[1] M.F Arlitt and C. Williamson, “Web server workload characterization, The search for invariants,” *In Proceedings of the ACM SIGMETRICS ’96 Conference*, pp. 126-137, Philadelphia, PA, April 1996.

[2] M.F Arlitt and C. Williamson, “Trace-Driven Simulation of Document caching Strategies for Internet Web Servers,” Dept. of Computer Science, University of Saskatchewan, 1996.

[3] P.Barford and M.Crovella, “Generating representative Web workloads for Network and Server Performance Evaluation,” *Presented in ACM SIGMETRICS International conference on Measurement and Modeling of Computer Systems*, pp.151-160, July 1998.

[4] Tim Bray, “Measuring the Web,” *In proceedings of fifth International World Wide Web conference*, Paris, France, 1996.

[5] J. Challenger, P. Dantzing and A. Iyengar, “A scalable and highly available system for serving dynamic data at frequently accessed Web sites,” IBM Research, T. J. Watson Research Center, 1998.

[6] M.E. Crovella and A. Bestavros, “Self-similarity in World Wide Web traffic: Evidence and possible causes,” *IEEE/ACM Transaction on Networking*, Vol. 5, No. 6, pp. 835 – 846, December 1997.

[7] M. Crovella and L Lipsky, “Long-lasting transient condition in simulations with heavy tailed workloads,” *Proceedings of the 1997 Winter Simulation Conference*, 1997.

[8] M. E. Crovella and M. S. Taqqu, “Estimating the heavy tailed index from Scaling Properties,” *Methodology and Computing in Applied Probability*, Vol. 1 No.1, 1999.

[9] S. Deng, “Empirical model of WWW document arrivals at access link,” In *proceedings of the 1996 IEEE International Conference on Communication*, pp. 1797-1802, June 1996.

[10] P. Embrechts, C. Klueppelberg and T. Mikosch, *Modelling extremal events for insurance and finance*, pp. 330-34, Springer, New York, 1997.

[11] V. Holmedahl and B. Smith and T. Yang, “Cooperative Caching of Dynamic Contents on a Distributed Web Server,” In P*roceeding. Of 7th IEEE International Symposium on High Performance Distributed Computing (HPDC-7),* pp.243-250, Chicago, IL USA, July *1998.*

[12] B. Krishnamurthy and J Rexford, “ Software Issues in Characterizing Web server logs,” *W3C Web Characterization Group Workshop,* November 1998.

[13] Related address for 100 hot.com is: http://www.100hot.com/home.chtml

[14] Related address for NLANR proxy cache is: ftp://ircache.nlanr.net/Traces

[15] S. Kotz, N. Johnson, *Encyclopedia of Statistical Sciences*, Volume 3, pp. 598-599, Wiley, New York, 1982-1988.

[16] H.F. Nielsen, J. Getty, A. Baird-Smith, E. Prud'hommeaux, H. W. Lie and Chris Lilley, "Network performance effects of HTTP/1.1, CSS1 and PNG," *Proceedings of SIGCOMM '97*, Cannes, France, September 1997.

[17] Microsoft Corporation, http://www.microsoft.com.

[18] C. E. Willes and M. Mikhailov, "Toward a better understanding of Web resources and server responses for improved caching", *Proceedings of the Eight International World Wide Web Conference*, pp. 153-167, Toronto, Canada, May 1999.

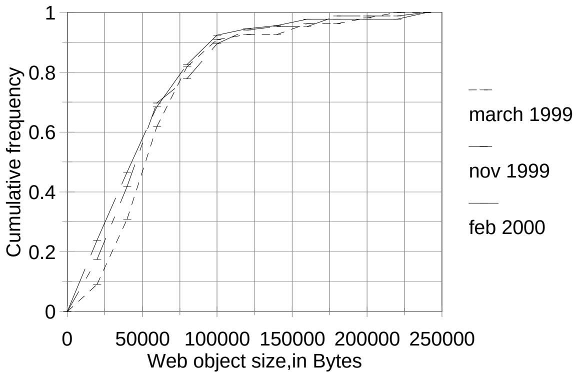

Figure 1: Web object sizes for popular data sets

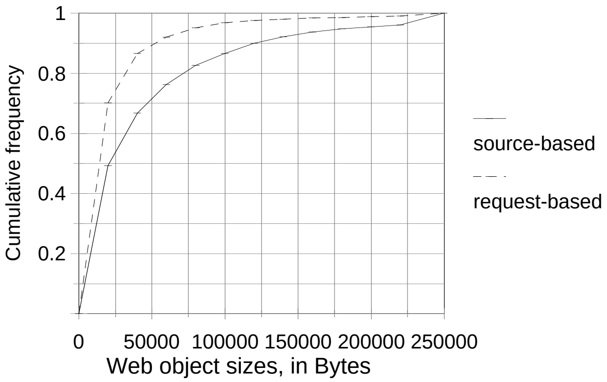

Figure 2: Web object sizes for log data sets

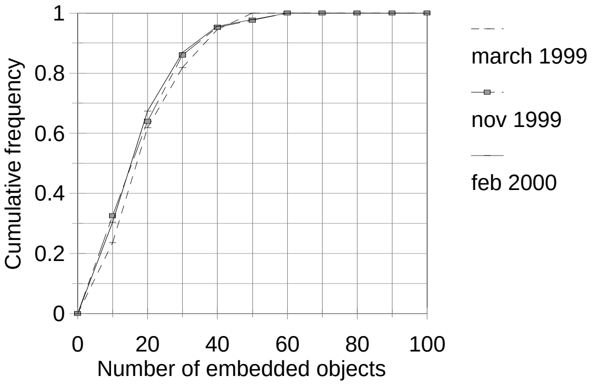

Figure 3: Number of embedded objects for popular sets

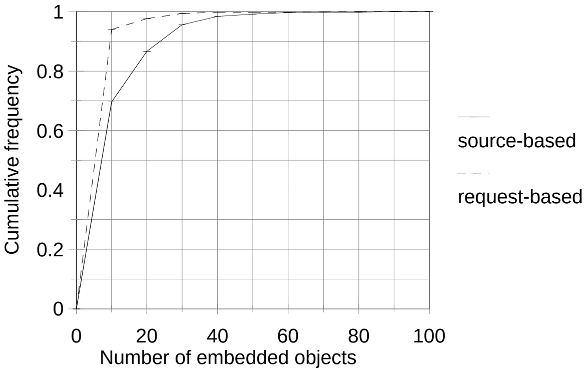

Figure 4:Number of embedded objects for log data sets

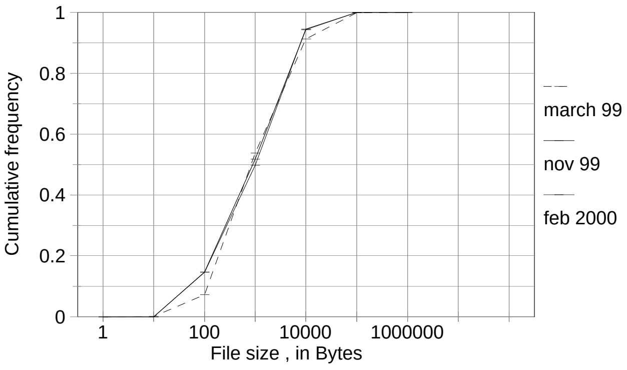

Figure 5: Cumulative frequency of file sizes for popular data sets

Table 1: Statistics on popular data sets.

Characteristic	Web object sizes (bytes)			No. of embedded objects			Web file sizes (bytes)		
Popular data set	March 1999	Nov 1999	Feb 2000	March 1999	Nov 1999	Feb 2000	March 1999	Nov 1999	Feb 2000
Mean	60061	54686	50818	17.8	17.5	16.1	3256	2563	2772
Median	47454	46413	41712	16	15	14	877	1013	947
Variance	1.7E+9	1.9E+9	2.1E+9	126.3	152.2	144.7	4.9E+7	2E+7	2.6E+7
Maximum	217839	285252	283303	45	59	57	101333	52193	49679
Coefficient of variation	0.86	0.80	0.91	0.63	0.71	0.75	2.15	1.74	1.9

Table 2: Statistics on popular Audio/Video files

Mean (bytes)	2976110
Median (bytes)	2883584
Mode (bytes)	4613734
Variance	4.96E+12
Minimum (bytes)	26624
Maximum (bytes)	10171187
Coefficient of Variation	0.75

Table 3: Summarized characteristics of Web objects for Web documents

Characteristic of Web object	**Popular Web documents**	**Regular Web documents**
Distribution of Web object sizes	heavy tailed	no data in previous work heavy tailed in log data
Web object size statistics	90% less than 100KB	no data in previous work 90% less than 100KB in log data
Distribution of number of embedded objects	not heavy tailed	not heavy tailed [6, 3] not heavy tailed in log data
Mean number of embedded objects	16.14 to 17.78 (see Table 2)	not reported
File size distribution	heavy tailed	heavy tailed [1, 6, 3]
File size statistics	90% of the files are less than 10,000 bytes	most of the files are in the range of 100 – 100,000 bytes [1]