Object Packaging - Web Response Time Reduction For Slow and Busy Web Servers

Hiroshi Fujinoki

Department of Computer Science
Southern Illinois University Edwardsville
Edwardsville, Illinois 62026-1656, USA
E-mail: hfujino@siue.edu

Kiran K. Gollamudi

Department of Electrical Engineering
Southern Illinois University Edwardsville
Edwardsville, Illinois 62026-1656, USA
E-mail: kgoramudi@siue.edu

Abstract

In this paper, an ongoing research activity to reduce response time in web file transmissions is presented. A new transmission technique, object packaging, is proposed for reduced response time for web browsing. Object packaging aims to reduce overhead not only at routers on a transmission path but at a transmitting web serve for better response time. The object packaging intends to address two different issues in web file transmissions; 1) disk access overheads and 2) protocol processing overheads. Experimental study showed that the file transmissions by object packaging reduced response time, number of transferred bytes and the packets by 34.7, 40.1 and 7.1%, respectively, for files with their average file size being 10K bytes, which is the average file size in the web traffic in the current Internet.

1. Introduction

In the current Internet, web traffic occupies the largest portion of the Internet traffic. Since the advent of Mosaic in 1993, the traffic on the Internet has been increasing at an exponential pace and this trend still exists even these days [1]. As a result, one of the most serious problems web users are experiencing is long response time and transmission delay due to high web traffic load [2].

Since the World Wide Web (called www hereafter) is based on the client-server model, delays, which are the causes of longer response time, are inserted at various places. However, major sources of delay are classified in the following two groups: 1) delay due to overhead at the server side and 2) delay during propagation in the transmission network. The delay due to server-side overhead consists of operating system overhead, such as file access latency, memory copies and packet processing overhead by a network protocol. The operating system overhead will be a problem, 1) if a server is overwhelmed by excess requests (i.e., a busy server) or 2) if a server is driven by a slow CPU or does not have enough memory (i.e., a slow server). Overhead for memory copies and packet processing such as CRC checksum calculation will be relatively high for small packets. The second group, the delay during propagation in the transmission network, consists of queuing delay and routing table look-ups at intermediate routers, and delay due to packet re-transmissions due to packet losses from network congestion. In this study, the delay due to packet losses during transmission is not considered.

2. Related Work

In order to reduce response time in web file transmission, various solutions have been proposed. For the overhead in operating system, Ousterhout [3] and Druschel [4] analyzed major factors in the overhead in operating systems and found that the memory copy and disk I/O are the largest overheads that limit throughput for file server processes. Markatos suggested that reducing the number of memory copies would improve TCP/IP performance for HTTP [5]. Busari suggested that the majority of web traffic consists of requests to many small files and only a few large files exist [6] (average file size is about 10K bytes in the web [1]). Since disk access overhead dominates the operating system overhead, server-side caching is one of the techniques to reduce the delay due to disk access. However, many of the requested files (50 to 70%) are referenced only once [6], implying the limitation of server-side caching to improve response time due to operating system overhead [7, 8, 9, 10].

3. File Transmission by Object Packaging

The major premise of web file transmissions by object packages is based on the fact that most homepages consist of many small files (files less than 16K bytes). Since the majority of the server-side overhead comes from operating system overhead to retrieve requested files from a local disk, the server-side overhead can be reduced if the number of requested files is reduced. Moreover, if we can reduce the number of requested files, the number of packets to be transferred in a network will be reduced, by avoiding packet fragmentation.

Based on the facts described in the previous paragraph, the web file transmission by object packaging is proposed. An object package is a collection of web files in a homepage packed in an uncompressed archive file. Multiple files, such as HTML text and image files, will be packed in an object package file for efficient transmission. Those files in a homepage are sequentially packed into an object package file without compression to prevent decompression overhead at the receiver side. At the receiving side, a web browser should unpack the files in an object package file. Object packages should be recognized by a specific file extension. The format of an object package is shown in Figure 1.

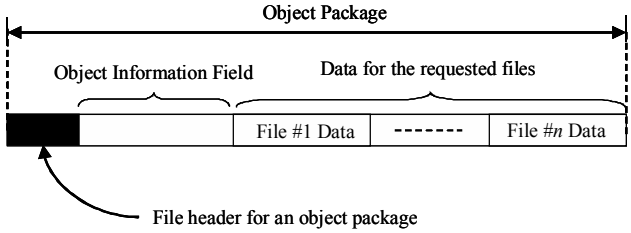

Figure 1 — Major components of the object package

The first element in an object package is the object information field. The object information field contains the information of the packed files. The data field contains the contents of the requested files. A receiver reconstructs the requested files using the information in the object information field and the contents (from data field) of the requested files. The object information field consists of four subfields (Figure 2).

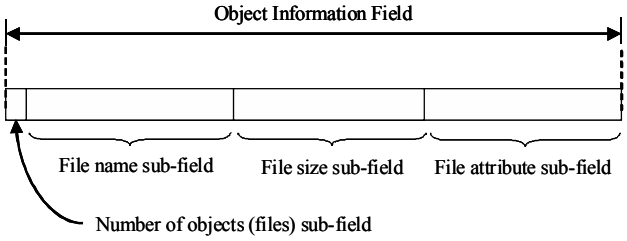

Figure 2 — Elements in object information field

The first subfield is the number of packed files. The second is the filename subfield, which is a collection of the names of the contained files. The third subfield contains the file sizes of the packed files. The last subfield indicates the attributes of each file such as binary, text, or executable. The first file (File #1) will be reconstructed based on the first element in the file name, the file size and the file attribute subfields.

4. Simulation Experiments

Simulation experiments were performed to evaluate the performance of web file transmission by object packaging. The web server machine and a client machine are connected through a hub. Transmission rate is 10 Mbps for all of the links. All of the requested files are initially stored in the local hard drive at the server machine. To capture the network traffic between the server and the client, another machine, called the packet monitor, is connected to the hub. The testbed is shown in Figure 3.

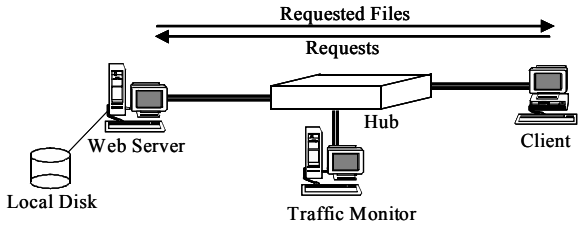

Figure 3 - The testbed for simulation experiments

Using the testbed, we measured the following factors for files transferred by the existing method and object packages: 1) number of actually transferred bytes (including packet header), 2) number of transferred packets, and 3) the response time. The response time was measured by the traffic monitor and is defined to be the time between the first packet transferred from the client to the web server (i.e., the TCP SYNC message) and the last packet transferred from the web server to the client (i.e., the last packet for the requested file).

Tables 1 through 3 show the results of the experiments. Table 1 shows the comparisons of the existing file transmission and the object packaging in the number of bytes actually transferred. Table 2 shows the number of packets actually transferred and Table 3 shows the average response time in seconds. Figure 4 shows the performance of the object packaging relative to the existing transmission method. For small files (average file size of 10K bytes or less), the object packaging demonstrated a significant reduction in the number of transferred bytes and packets, and response time. When the average file size is 1K bytes, 43.7%, 87.8% and 80.7% reduction were made for the transferred bytes, transferred packets and the response time.

Table 1 — Average transferred bytes

	Existing Method	Object Package
1K	188362.3	106057.1
4K	491916.1	409423.9
10K	1094350.5	1016420.7

Table 2 — Average number of packets

	Existing Method	Object Package
1K	944.2	115.3
4K	1189.2	420.2
10K	1720.4	1015.6

Table 3 — Average response time (in seconds)

	Existing Method	Object Package
1K	0.696	0.134
4K	0.927	0.403
10K	1.531	0.999

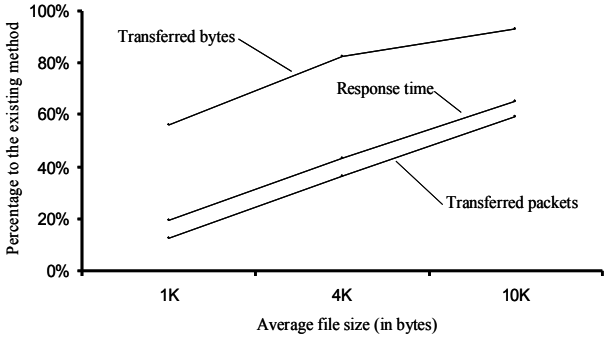

Figure 4 — Results for different file sizes

Figure 5 shows the results for different numbers of transferred files. The results showed that the relative performance of the object packaging would not be dependent on the number of transferred files as long as at least more than ten files are transferred, meaning that the advantages of the object packaging will be effective even for transferring a small number of files.

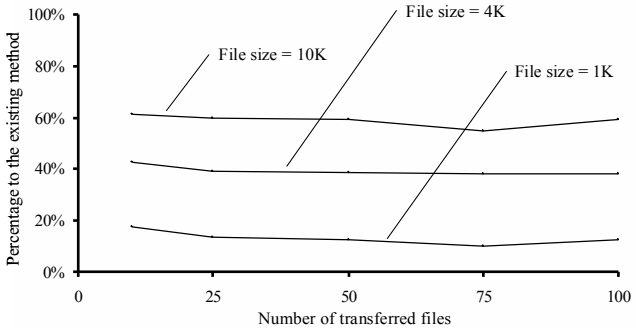

Figure 5 — Results for different number of files

5. Conclusions and Future Work

In this paper, an ongoing research activity for efficient web file transmissions, called multiple file transmission by object package, is proposed. The technique does not require any modification of an operating system at the server side nor transmission protocol at routers. Experiments show that object packaging is efficient in reducing response time. It was observed that the object packaging reduced response time, number of transferred bytes and the packets by 34.7, 40.1 and 7.1%, respectively, for files with the average file size being 10K bytes, which is the average file size in the web in the current Internet. Currently, to observe the scalability of the object packaging transmission, experiments are being performed.

6. References

[1] M. Arlitt and C. Williamson, "Web Server Workload Characterization: The Search for Invariants," *Proceedings of the 1996 ACM SIGMETRICS Conference on the Measurement and Modeling of Computer Systems*, May 1996, pp. 126-137.

[2] GVU's WWW User Surveys, *Georgia Institute of Technology* URL*: http://www.gvu.gatech.edu/user_surveys*

[3] J. Ousterhout, "Why Aren't Operating Systems Getting Faster As Hardware?," *Proceedings of Summer 1990 USENIX Conference*, June 1990, pp. 247-256.

[4] P. Druschel, "Operating System Support for High-Speed Networking," *Communications of the ACM*, vol. 39, no. 2, September 1996, pp. 41-51.

[5] P. Markatos, "Speeding-up TCP/IP: Faster Processors Are not Enough," *Proceedings of the 21st IEEE International Performance, Computing, and Communications Conference*, April 2002, pp. 341-345.

[6] M. Busari and C. Williamson, "On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics," *Proceedings of IEEE INFOCOM*, April 2001, pp. 1225-1234.

[7] J. Dilley, "The Effect of Consistency on Cache Response Time," *IEEE Network*, vol. 14, no. 3, May/June 2000, pp. 24-28.

[8] S. Glassman, "A caching relay for the World-Wide Web," *Computer Networks and ISDN Systems*, vol. 27, no. 2, October 1994, pp. 165-173.

[9] D. Lee, "Pre-Fetch Document Caching to Improve World-Wide Web User Response Time," Master's Thesis. Virginia Polytechnic Institute and State University, March 1996.

[10] J. Mogul, "Squeezing More Bits Out of HTTP Caches," *IEEE Network*, vol. 14, no.3, May/June 2000, pp. 6-14.