Preliminary Measurements on the Effect of Server Adaptation for Web Content Delivery

Balachander Krishnamurthy, Craig Wills, Yin Zhang

I. Introduction

A Web client experiences poor performance due to low bandwidth, high latency, network congestion, etc. A server can select a lower quality version of the resource or alter the manner of content delivery to improve performance. We present early measurement results on the actual latency reduction for a wide class of real and geographically dispersed set of clients. Earlier research work in compression, delta encoding, use of content distribution networks (CDNs), etc. has examined Web performance via the lens of individual improvements in reducing user-perceived latency or load on servers. They use different methodologies, workloads, and validation techniques. We examine multiple performance related factors in a single unified framework, a set of server actions to improve performance, and use a canonical set of container documents with various distributions of embedded objects in terms of number and size. Our work can be applied by a variety of sites to test the potential improvement of clients that visit them.

Our active measurement testbed consists of clients with different connectivity sending requests to a few Web servers under our control, helping us to examine performance components in an automated fashion. By downloading a canonical container document set via clients with different connectivity capabilities, we can measure the actual improvement as a result of various server actions. Actions include altering the content (server choosing a reduced version for poorer clients by including fewer or thinner embedded objects), using a CDN when round-trip latency between client and server is high, altering how content is delivered (by compressing, or bundling embedded objects to avoid multiple retrievals, or combining these two methods), and by varying policy associated with maintaining persistent connections (keep connection open longer with poor clients).

Krishnamurthy and Zhang are with AT&T Labs–Research, Florham Park, NJ, USA. Wills is with Worcester Polytechnic Institute, Worcester, MA, USA.email:{bala,yzhang}@research.att.com, cew@cs.wpi.edu. Contact author: Balachander Krishnamurthy, Fax: 973-360-8077, 180 Park Avenue, Florham Park, NJ 07932.

II. Methodology

We want to investigate actions that a Web server can take to reduce download time for a client. While dynamic generation of Web content can take a nontrivial amount of time, the generation is under control of the server and can be improved independent of content delivery. The frequency and amount that content changes can significantly affect the usefulness of caching and delta encoding, but these techniques are only relevant for repeat accesses by a client or cluster of clients for a page.

For now, we focus on characterizing pages based on the amount of content of a page because this affects the first access by a client for the page. We examine the number of bytes in the container object, the number of embedded objects and the total number of bytes for the embedded objects and investigate which server actions are the most effective in reducing download times for different combinations of characteristics that a page may have. We identified Web content that covered the "space" of these characteristics and populated a canonical test site with realistic content known to be requested by clients. The site was then used for retrieval of content in the context of different server actions.

We used recent proxy logs from a large manufacturing company with over 100,000 users, examined requests to the container object of a page by looking for HTML URLs, and selected the 1000 most popular pages. In April 2002 we downloaded each container object and embedded objects (frames, layers, cascading style sheets, Javascript code and images) to determine the size of these objects. Objects referenced as a result of executing embedded Javascript code were not considered. 641 URLs containing one or more embedded objects were successfully retrieved and using 33% and 67% percentile values we created a small, medium and large value range for each characteristic. Using these three ranges for each of the three characteristics defines a total of 27 "buckets" for the classification of an individual page. The cut off for container bytes in small, medium and large were less than

12K, less than 30K bytes, and more than 30K bytes respectively. Similarly, for the number of embedded objects it was less than 7, 22, and more than 22 and for embedded bytes 20K, 55K, and more than 55K bytes.

We determined the number of pages that fell in each bucket with the percentages available in [2]. We also looked at the home pages of 131 popular [3] Web sites using the same bucket ranges, as a comparison. These pages have a larger number of embedded objects and bytes than the popular pages from the company proxy log. This distribution simply indicates that the effect of server actions for more embedded content is of greater interest for popular site home pages.

We defined the ranges primarily to identify pages that spanned the space of all possible characteristics. We selected two representative pages from each bucket of the proxy log pages. In buckets containing many pages we tried to select two pages that were representative of characteristics within the bucket. In all, we selected 44 pages (not all buckets contained two pages) to cover the space of characteristics and downloaded them to a test site using *wget*. *wget* localizes links to embedded and traversal links, although it does not identify objects such as style sheets and Javascript objects and so these were separately downloaded. For redirection responses, URLs in the container object were changed to match the downloaded name.

Additional objects were created in preparation for testing the various server actions: compressed version of each container object using *gzip*, single bundled object with the embedded objects for each page, and a separate compressed bundle object. For offloading embedded content from a server to a CDN, we crawled the content of a Web site known to use a major CDN[1] and cataloged a large number of objects along with sizes served by this CDN. We then matched each object at our test site with a similar size CDN-served object. We installed the test site on unloaded servers on both coasts of the U.S. We then used *httperf* [4] from six other client sites to make automated retrievals to each test server and CDN site for testing of the various server actions. See [2] for details of the results; they are summarized below.

III. Conclusions and Ongoing Work

This is the first study that we are aware to look at the impact of server actions for a variety of content and client conditions where each action is measured on a common platform and cumulative effect of two or more actions can be evaluated. High volume websites can benefit from our results by examining their content mix to see how different actions will benefit their clients. The summary of our experimental results show that:

- Compression of HTML content is not universally useful. We did not find that compression had a significant effect on reducing response time for well-connected clients. Compression is an effective action when the client is bandwidth-constrained.
- The CDN was useful for improving performance of well-connected clients, but not so for bandwidth-constrained clients even when it provided a lower RTT for clients. This result extends what we found in [1]. Note that these comparisons were made with relatively unloaded servers.
- Persistent connections with serialized requests do not provide a significant performance improvement under a wide variety of client/content conditions.
- Persistent connections when combined with pipelining are only significant for high bandwidth clients.
- Bundling content, like pipelining, has some use particularly for better connected clients. Using CDNs to serve bundles is also a good idea for well-connected clients. Compressed bundles can have a significant effect for all types of clients.
- In terms of lossy actions, removing embedded objects has a significant effect in all cases. However, reducing the quality of embedded objects without reducing the number does not yield a significant improvement under most circumstances.
- Client connectivity, not latency, matters for determining which actions have significant performance effects. Our results are consistent for clients with similar connectivity despite large variations in latency.

We are examining the cost to the clients for dealing with modified content sent by the server, as well as costs of unbundling and other actions. We are also looking at other possible server actions and policies regarding cachability of objects. An implementation with a modified Apache Web server that characterizes clients as per their connectivity and takes an appropriate action is under way.

[1] We did not use the AT&T CDN to avoid appearance of bias.

References

[1] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. On the use and performance of content distribution networks. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, November 2001.

[2] Balachander Krishnamurthy, Craig Wills, and Yin Zhang. Preliminary measurements on the effect of server adaptation for web content delivery. Technical Report TD-59VNB8, AT&T Labs – Research, April 2002. `http://www.research.att.com/~yzhang/papers/spinach-td02.pdf`.

[3] Media metrix, March 2002. `www.mediametrix.com`.

[4] D. Mosberger and T. Jin. httperf—a tool for measuring web server performance. In *Proceedings of WISP '98, Madison, WI*, June 1998.