Organization-Based Analysis of Web-Object Sharing and Caching

(Extended Abstract)

Alec Wolman, Geoff Voelker, Nitin Sharma, Neal Cardwell,Molly Brown, Tashana Landray, Denise Pinnel, Anna Karlin, Henry Levy

Department of Computer Science and Engineering
University of Washington

Abstract

Crucial to performance-enhancing mechanisms in the Internet are repeated requests of specific web documents by multiple clients. However, little is known about patterns of shared document access, particularly from diverse client populations. The principal goal of this paper is to examine the sharing of web documents from an *organizational* point of view. An organizational analysis of sharing is important, because caching is often performed on an organizational basis; i.e., proxies are typically placed in front of large and small companies, universities, departments, and so on. Unfortunately, simultaneous multi-organizational traces do not currently exist and are difficult to obtain in practice.

The goal of this paper is to explore the extent of document sharing (1) among clients within single organizations, and (2) among clients across different organizations. To perform the study, we use a large university as a model of a diverse collection of organizations. Within our university, we have traced all external web requests and responses, anonymizing the data but preserving organizational membership information. This permits us to analyze both inter- and intra-organization document sharing and to test whether organization membership is significant. As well, we characterize a number of parameters of our data, including basic object characteristics, object cacheability, and server distributions.

1 Introduction

The need to understand web behavior and performance has led to a large number of studies, aimed in particular at classifying web document *characteristics* [10, 19, 12, 15, 11]. In contrast, the principal goal of this study is to evaluate document *sharing behavior* on the web, both *within* organizations and *between* organizations. By document sharing, we mean access to the same web documents by different clients. Sharing behavior has obvious implications for performance, particularly with respect to the effectiveness of proxy caching (e.g., [13, 7, 18, 24, 8, 16]). Proxy caches are often located at organizational boundaries and improve performance only if many documents are shared by many clients. Therefore, an understanding of sharing gives us added insight into potential performance-enhancing mechanisms and alternative caching structures.

An analysis of document sharing within an organization is straightforward and can help predict the benefits of an organizational proxy cache [12]. Studying sharing across multiple organizations is much more difficult, however. Tracing of the entire web is obviously not achievable, but even simultaneous traces of multiple organizations do not currently exist. In addition, the requirement of most organizations for anonymization of URLs and IP addresses, along with the different dates of data capture, makes correlation of separate traces challenging, if not impossible.

In this study, we use The University of Washington (UW) as a basis for modeling intra- and inter-organizational web-object sharing. The UW is the largest university in the northwest part of the U.S., with a population of over 50,000 people, including 35,000 students, 10,000 full-time staff, and 5,000 faculty. The university has a large communications infrastructure, consisting of thousands of computers connected via both high-speed networks and modems. Together, this community generates a workload of more than 10,100 university-external web requests per minute at peak periods.

As with other universities, UW is organized into many colleges, departments, and programs, each with its own disparate administrative, academic, or research focus. For example, the UW includes museums of art and natural history, medical and dental schools, libraries, administrative organizations, and of course academic departments, such as music, Scandinavian languages, and computer science. What do such diverse organizations have in common with respect to their web access requests? To answer this question, we have traced all UW-external web requests; we anonymize the data in such a way as to identify requests (and associated responses) with the 170 or so independent organizations from which they were issued. This permits us to study organization-specific document access and sharing behavior. We have collected a number of traces during the period from October 1998 through the present. In general, all of our traces show the same basic patterns. The results in this paper are based on a representative one-week trace taken in mid-January 1999, and therefore show the very latest characteristics of modern web traffic.

The paper is organized as follows. The next section provides a brief description of related work. In Section 3 we describe our trace-capture methodology. Section 4 contains a high-level description of the workload we traced. Section 5 focuses on organization-based statistics and also provides inter- and intra-organization sharing analysis. In Section 6 we discuss cacheability of documents, and reasons why documents are not cacheable. Finally, Section 7 summarizes our study and its results.

2 Previous Work

Numerous recent studies of web traffic have been performed. These studies include analyses of web access traces from the perspective of browsers [10, 19], proxies [9, 2, 17, 12, 15, 11, 22, 5, 6, 4, 14], and servers [21, 1, 3]. The earlier tracing studies were rather limited in request rate, number of requests, and diversity of population. The most recent tracing studies have been larger and generally more diverse. In addition to static analysis, some studies have also used trace-driven cache simulation to characterize the locality and sharing properties of these very large traces [2, 17, 12, 15, 5, 14], and to study the effects of cookies, aborted connections, and persistent connections on the performance of proxy caching [5, 14].

In this paper, we expand on these previous research efforts. Our focus is on sharing and cacheability; however we can also compare our current HTTP traffic characteristics to earlier studies, showing how the web workload has changed. Our work is based on the most recent data from a large diverse population. More important, we preserve enough information so that we can analyze requests with respect to inter-organization and intra-organization document sharing.

3 Measurement Methodology

UW connects to its Internet Service Providers via two border routers; one router handles primarily outbound traffic and the other inbound traffic. These two routers are fully connected to four 100-megabit Ethernet switches distributed across the campus. Each switch has a monitoring port that is used to send copies of the incoming and outgoing packets to our monitoring host, a 500 MHz Digital Alpha workstation running Digital Unix V4.0. We designed and implemented the tracing software used to produce the data in this study. This software installs a kernel packet filter [20] to deliver all TCP packets from the network interfaces to the user-level monitoring process, which analyzes the packets and produces a trace.

We use an anonymization approach that protects privacy but preserves some address locality information. For internal addresses, we classify the IP address based on its "organization" membership. An organization is a set of university IP addresses that forms an administrative entity; an organization may include multiple subnets. Once the organization identifiers are assigned, both the IP address and the organization identifier are anonymized. Furthermore, some bits of information in the IP address are destroyed before anonymization in order to make the encryption more secure.

For external addresses, we anonymize each octet of the server IP address separately. For our purposes, two servers are near each other if they share most or all of the Internet path between them and the university. We consider two servers to be on the same subnet when the first three octects of their IP addresses match. Given the use of classless routing in the Inter-

HTTP Transactions (Requests)	43.4 million
Objects	12.3 million
Clients	21,203
Servers	234,115
Total Bytes	369 GB
Average requests/minute	4,300
Peak requests/minute	10,100

Table 1: Overall trace statistics.

net, this scheme will not provide 100% accuracy, but for large organizations we expect that this assumption will be overly conservative rather than overly aggressive.

All of our results are based on the entire trace collected from Tuesday January 12th through Tuesday January 19th, 1999, except for the organization-based sharing results in Section 5, which are from the first one-day period of our trace (the limitation is due to the memory requirements of the sharing analysis).

4 High-Level Data Characteristics

Table 1 shows the basic data characteristics. As the table shows, our trace software saw the transfer of 369 gigabytes of data in response packets, requested from about 21,000 client addresses, and returned from 234,000 servers. It is interesting that, compared to the commonly-used 1996 DEC trace (analyzed, e.g., in [12]), which had a similar client population, we saw twice as many requests in one week as DEC saw in 3 weeks. These requests and corresponding response and close events follow the typical diurnal cycle, with a minimum of 360 requests per minute (at 5 AM) and a peak of 10,100 requests per minute (at 4 PM).

The top content types we saw, by count, were image/gif (35.9%), text/html (32.1%), No Content Type (17.5%), and image/jpeg (10.5%), with all the rest of the content types at significantly lower numbers. The No Content Type traffic, which accounts for 18% of the responses, consists primarily of short control messages. The largest percentage of bytes transferred is accounted for by text/html with 31%. Types image/gif and image/jpeg are 19% and 18% of bytes transferred, respectively. The remaining content types account for decreasing numbers of bytes with a heavy-tailed distribution.

Another type that accounts for significant traffic, which does not appear in the table, is audio/video. Much of this traffic is streamed through an out-of-band channel between the audio/video player and the server, rather than through HTTP, and therefore our trace does not contain detailed audio information. We have extended our tracing software to analyze two kinds of external audio/video traffic. One extension handles port 7070 traffic (used by Real Networks audio/video player), and the other handles RTSP traffic. Our analysis code was only able to analyze about 30% of the above connections, and for those 30% we saw a total of over 61 GB of audio/video traffic coming into the UW from the Internet. This is the equivalent of 17% of all HTTP traffic during that same period. The audio/video objects themselves are very large, with an average size of 2.3 MB. The largest single transfer seen was 1.9 GB. We believe that the quantity of audio and video is signaling a new trend; e.g., in the data collected by [15] and [11], audio traffic does not appear. [1]

We were also curious about the HTTP protocol versions currently in use. The majority of requests in our trace (64%) are made using HTTP 1.0, but the majority of responses use HTTP 1.1 (57%). In terms of bytes transferred, the majority of bytes (56%) are returned using HTTP 1.0.

These statistics simply serve to provide some background about the general nature of the trace data, in order to set the context for the analysis in the next two sections.

5 Analysis of Document Sharing

This section presents and analyzes our trace data, focusing on document sharing. As previously stated, our intention is to use the university organizations as a simple model of independent organizations in the Internet. Our goal is to answer several key questions with respect to web-document sharing, for example:

1. How much object sharing occurs between different organizations?
2. What types of objects are shared?
3. How are objects shared in time?

[1] One should not assume that the 70% of the connections that we were unable to analyze have similar characteristics, because it may be that the short connections are the ones that do not contain enough information in the control connection to perform the size analysis.

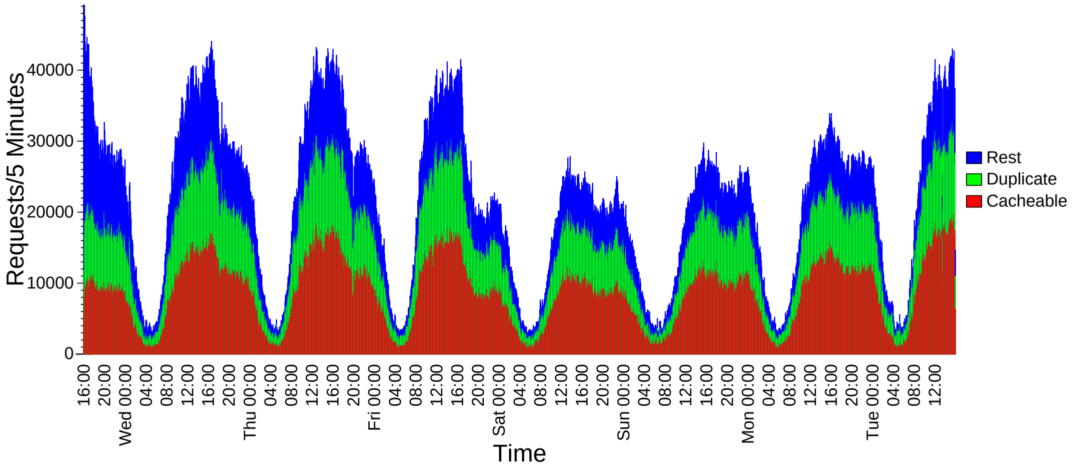


Figure 1: Requests broken down into initial, duplicate, and duplicate cacheable requests over time.

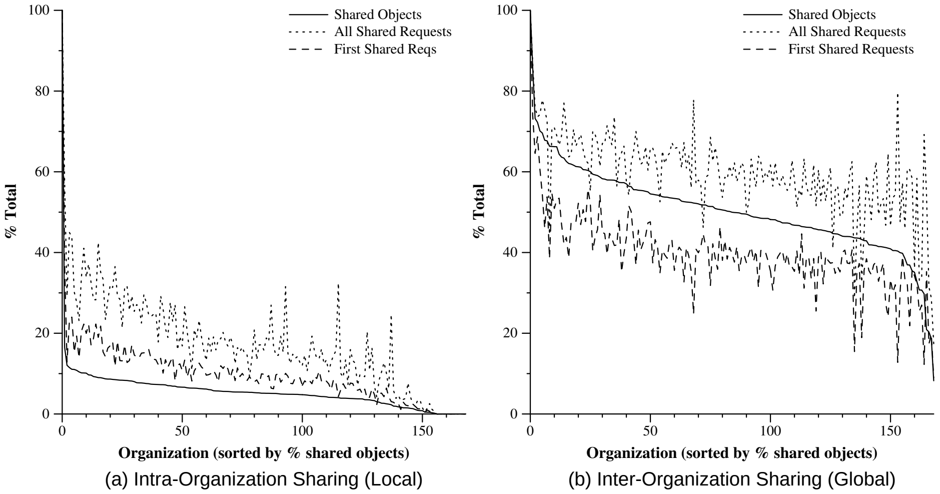


Figure 2: The left graph shows the fraction of objects and requests accessed by the organization that are shared by more than one client within the organization. The right graph shows the fraction of objects and requests accessed by the organization that are shared with at least one other organization.

4. Is membership in an organization a predictor of sharing behavior?
5. Are members of organizations more similar to each other than to members of different organizations, or do all clients behave more-or-less identically in their request behavior?

Figure 1 plots total web requests per 5 minute period over the one-week trace period. The shading of the graph divides the curve into three areas: the darkest portion shows the fraction of requests that are initial (first) requests to objects, while the lightest portion shows the subset that are duplicate (repeated) requests to documents. A request is considered a duplicate if it is to a document previously requested in the trace by any client. The medium color shows those requests that are both duplicate and cacheable, as we will discuss later.

Overall, the data shows that about 72% of requests are to objects previously requested in the trace. This matches fairly closely the results of Duska et al. on several large organizational traces [12]. The percentage of shared requests rises very slowly over time, as one might expect. From our one-week trace, we cannot yet see the peak; however, this analysis does not consider document timeouts or replacements, therefore the 72% is optimistic if used as a basis for prediction of cache behavior. Furthermore, we cannot tell from the figure how many of the requests to a shared object were duplicate requests from the same client; overall, we found that about 62% of the requests to shared documents were first requests by a client to those documents; 38% were repeat requests by clients.

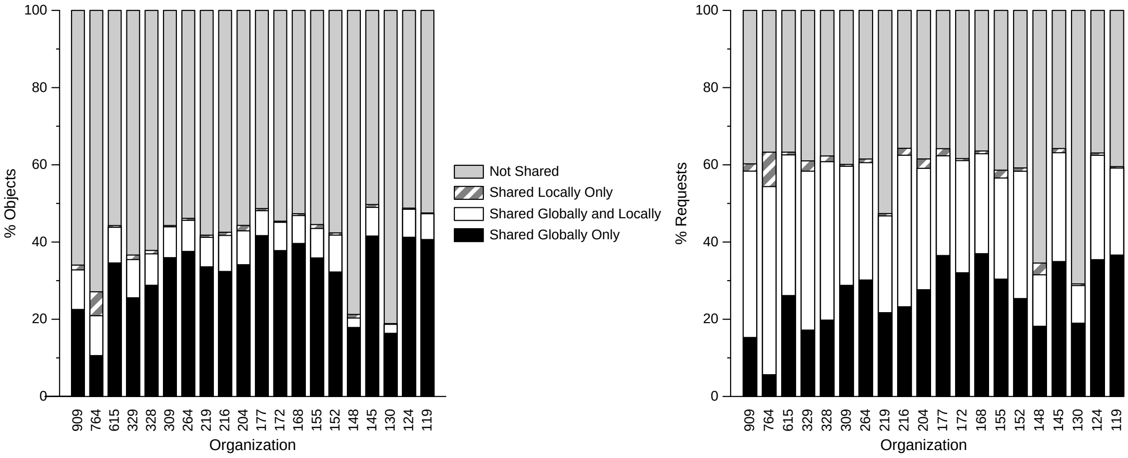
A key component of our data is the encoding of the organization number, which allows us to identify each client as belonging to one of the 169 active university organizations. These organizations include academic and administrative departments and programs, dormitories, and the university-wide modem pool. There are several very large organizations, with most somewhat smaller. The largest organization has 909 "anonymized" clients, the second largest organization is the modem pool with 764 clients, and the third largest organization has 615 clients.[2] The top 20 organizations all have more than 100 clients, as shown by the labels in Figure 3. Because of the way that client IP addresses are anonymized, we cannot uniquely identify an individual client, i.e., each anonymized client address could correspond to up to 4 separate clients. For this trace the ratio of "real" clients to "anonymized" clients measured by the low levels of our trace software is approximately 1.6; therefore, our 12,882 anonymized clients represent 21,203 true clients.

Using the organization data, we can analyze the amount of object sharing that occurs both within and between organizations. Figure 2a shows intra-organization (*local*) sharing from three perspectives. The solid line shows, for each organization, the percentage of all *objects* accessed by that organization that are locally shared (i.e., accessed by more than one organization member). The dotted line shows, for each organization, the percentage of all *requests* made by that organization that are to locally-shared objects. The dashed line shows, for each organization, the percentage of all requests made by that organization that are *first* requests by a given client to a locally-shared object. The organizations are ordered by decreasing locally-shared object percentage.

From this figure, we see that of all the objects accessed by members of an organization, only a small percentage are shared among organization members (5.1% on average). However, a much larger fraction of requests (16.8%) are to locally-shared objects, indicating that those shared objects are "hotter" than the non-shared objects. (Note that an object shared locally may also be shared globally; we show a more detailed breakdown of this distinction below.) From the distance between the dotted and dashed curves, we see that on average each anonymized client accesses each shared document several times.

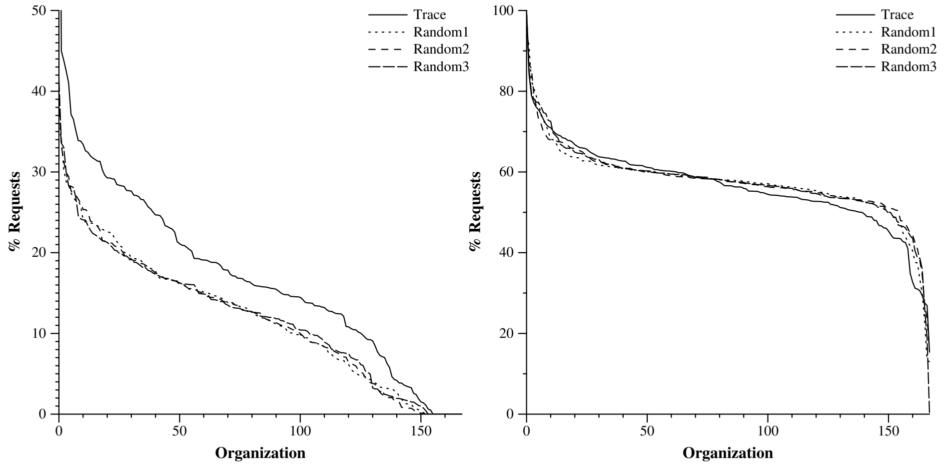
Figure 2b shows the inter-organization (*global*) sharing activity. Here the solid line shows the fraction of objects accessed by an organization that were also accessed by other organizations. Similarly, the dotted line shows the fraction of all organization requests to objects also accessed by other organizations. The dashed line shows the fraction of all organization requests that are first requests by a given client to a globally-shared object. We see that the solid line in Figure 2b is much higher than the corresponding line in Figure 2a, which indicates that there is more sharing with other organizations than within

[2]The modem pool is somewhat special, because multiple clients can login through a single IP address in the pool.



(a) Object Sharing (b) Request Sharing

Figure 3: Breakdown of objects (a) and requests (b) into the different categories of sharing, for the 20 largest organizations. The labels on the x-axis show the number of clients in each organization shown.



(a) Intra-Organization Sharing (Local) (b) Inter-Organization Sharing (Global)

Figure 4: Fraction of requests in the organization that are shared within this organization (a) and shared with at least one other organization (b), compared with three random client to organization assignments.

the organization. This is not surprising, since the overall client population is extremely large, so there is a much greater opportunity for sharing with clients in other organizations. For most of the organizations, more than half of the objects referenced are globally-shared objects. Similarly, there is a high percentage of object references that are to globally-shared objects.

A key question raised by these figures is whether the objects shared within an organization are the *same* set of objects that are shared across organizations. Figure 3a shows, for the 20 largest organizations, a breakdown of organization-accessed objects into various sharing categories: local only, global only, local and global, and not shared. Figure 3b shows the same breakdown by request. The graphs are ordered in decreasing organization size, with the organization size shown on the x-axis.

From Figure 3b, we see that the fraction of requests to shared objects is fairly flat across these organization sizes. As we would expect, the fraction that are shared globally-only rises somewhat with decreased organization size, while the fraction that are locally-shared decreases with decreasing organization size. That is, in general, the smaller the organization, the less organization-internal sharing, and the more global sharing. Looking at the white section of the bars in both figures, we see that the small percentage of objects that account for both local and global sharing are very hot, and account for a much greater fraction of the requests than the objects they represent. In contrast, the percentage of requests to objects shared locally-only is very small for these organizations.

To better understand the degree of object sharing, Figure 5 plots the number of objects (on the y-axis) that were shared by exactly x organizations. Most objects are accessed by only one organization, as shown by the steepness of the curve at $x = 1$. The graph also shows that there were 1000 objects accessed by 14 organizations and 100 objects accessed by 37 organizations.

A key question with respect to our sharing data is whether organization membership is significant. To answer this question, we randomly assigned clients to organizations, and compared the inter- and intra-organization sharing in the random assignments with the sharing seen in our trace analysis presented above. (The randomly-assigned organizations had the same sizes as the actual organizations.) Figure 4a plots the fraction of requests to locally-shared objects of the trace organizations and three randomly-assigned organizations. From the figure, we see that over all of the organizations, sharing is higher in the real organizations than in the randomly-assigned organizations. In other words, there is locality of references in organization membership. Figure 4b plots the fraction of requests to globally-shared objects for the trace and for the three random organizations. Here, as expected, there is no significant difference in the amount of global sharing between the real trace and the randomized organization assignment.

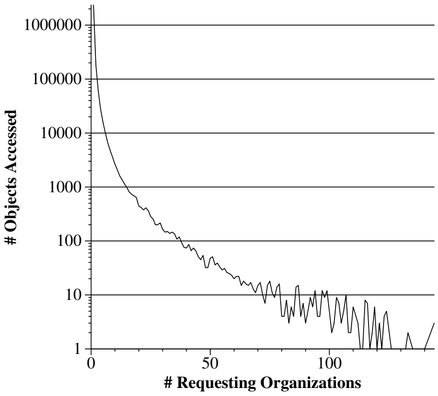


Figure 5: The number of objects accessed by a given number of organizations.

The organization-oriented data show that there is, in fact, significance to organization membership. Members of an organization are more likely to request the same documents than a set of clients of the same size chosen at random. However, the vast majority of the requests made are to objects that are *globally* shared. In addition, objects that are shared both locally within an organization and globally with other organizations are more likely to be requested by an organization member. This suggests that the most requested objects are universally popular.

Object and Server Popularity

For another aspect of sharing patterns we examine the servers that are being accessed and server proximity (i.e., which servers are close to each other in the network).

Figure 6 shows the cumulative distribution function of server popularity. This graph indicates that

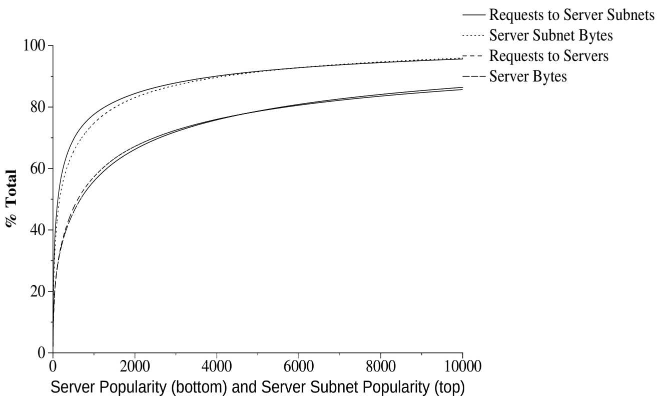


Figure 6: The cumulative distributions of server and server subnet popularity.

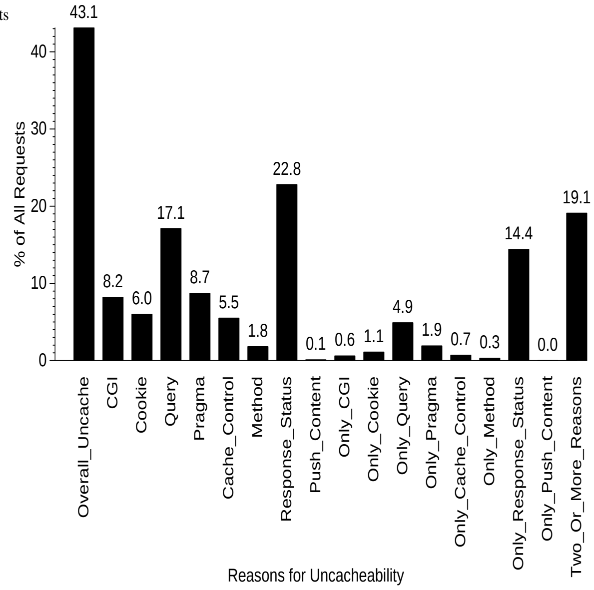


Figure 7: Reasons for uncacheability of HTTP transactions.

50% of the objects accessed and bytes transferred come from roughly the top 650 servers (out of a total of 234,000 servers accessed). Figure 6 also shows the cumulative distributions of server subnet popularity. A subnet is an organization of servers that share the same first 24 bits of their IP addresses. Such organizations of servers are typically mirrors of each other, or at least sit in a single server farm owned by a single company. We see that 50% of the objects come from about the top 120 server subnets; 20% come from the top 10 subnets.

6 Document Cacheability

This section examines cacheability of documents, giving us insight into the potential effectiveness of proxy caches in our environment. Web proxy caches are a key performance component of the WWW infrastructure; their objective is to improve performance through caching of documents requested more than once. Proxies typically live at the boundaries of an organization, caching documents for all clients within that organization.

In Figure 1 we saw a time-series graph of the percentage of duplicate requests (i.e., requests to a previously-accessed document) and cacheable requests in our trace. The cacheable requests are those made to documents that would be cached by a standard proxy cache, such as Squid [23]. From the figure, we see that in steady state, approximately 41% of the requests are duplicate and cacheable, placing an upper bound on the hit rate. The wide difference between the duplicate line and the cacheable line indicates that only about half of the duplicate requests (which could benefit from caching) are to objects that are cacheable.

Our cacheability analysis is based on the implementation of the Squid proxy cache. We examine the policies implemented by both Squid v1 and Squid v2. There are several reasons why a document may be uncacheable.

- *CGI* – The document was created by a CGI script or program and is not cached, because it is produced dynamically.
- *Cookie* – The response contains a set-cookie header. Squid v1 does not allow these responses to be cached, but Squid v2 does allow them to be cached.
- *Query* – The request is a query, i.e., the object name includes a question mark (“?”).
- *Pragma* – The response is explicitly marked uncacheable with a “Pragma: no-cache” header.
- *Cache-Control* – The response is explicitly marked uncacheable with the HTTP 1.1 Cache-Control header.
- *Method* – The request method is not “GET” or “HEAD”.
- *Response-Status* – The server response code does not allow the proxy to cache the response. For example, response code 302 (Moved Temporarily) cannot be cached when there is no explicit expiration date specified.

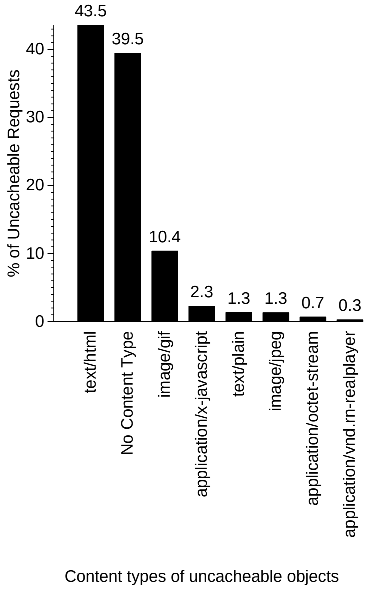


Figure 8: Breakdown by content-type of the uncacheable HTTP transactions.

- *Push-Content* – The content type "multipart/x-mixed-replace" is used by some servers to specify dynamic content.
- *Auth* – A future version of the trace software will detect requests that specify an Authorization header, as these cannot be cached.

Figure 7 shows a breakdown of all HTTP requests, detailing the percentage that are uncacheable for each of the reasons listed above. As the figure shows, 43% of the requests are uncacheable for one or more of the itemized reasons. Queries and Response Status are the two major reasons for uncacheability. Adding up the percentages for each reason sums to an amount greater than the overall uncacheability rate, showing that many documents are uncacheable for multiple reasons. The figure also shows, for each itemized reason, the percentage of HTTP requests that are uncacheable only due to that reason. Finally, the figure shows that 19% of web requests are uncacheable for two or more reasons. Figure 8 shows the most common content types for the uncacheable documents.

Figure 9a shows, for each organization, the percentage of objects (solid line) requested by the organization that are potentially cacheable. The dotted line shows, for each organization, the percentage of requests whose responses are cacheable. The Figure shows that the percentage of cacheable requests is significantly lower than the percentage of cacheable objects. The percentage of cacheable requests gives an upper bound on the hit rate each organization would see with an organization-local proxy cache.

Figure 9b shows, for each organization, the percentage of objects (solid line) and the percentage of requests (dotted line) that are both cacheable and shared with some other organization. The dashed line shows, for each organization, the percentage of requests that are first requests by a client to a trace-shared and cacheable object. These numbers give us an upper bound on the difference between the hit rate each organization would see with a global proxy cache for the entire university compared to having only an organization-local proxy cache. Thus, the marginal hit rate benefit of being able to access data cached by another organization is fairly small, even though the overall population size is more than 169 times as big as the average organization size. The reason is that only the first miss to an object within an organization would be converted from a miss to a hit, given a global proxy cache.

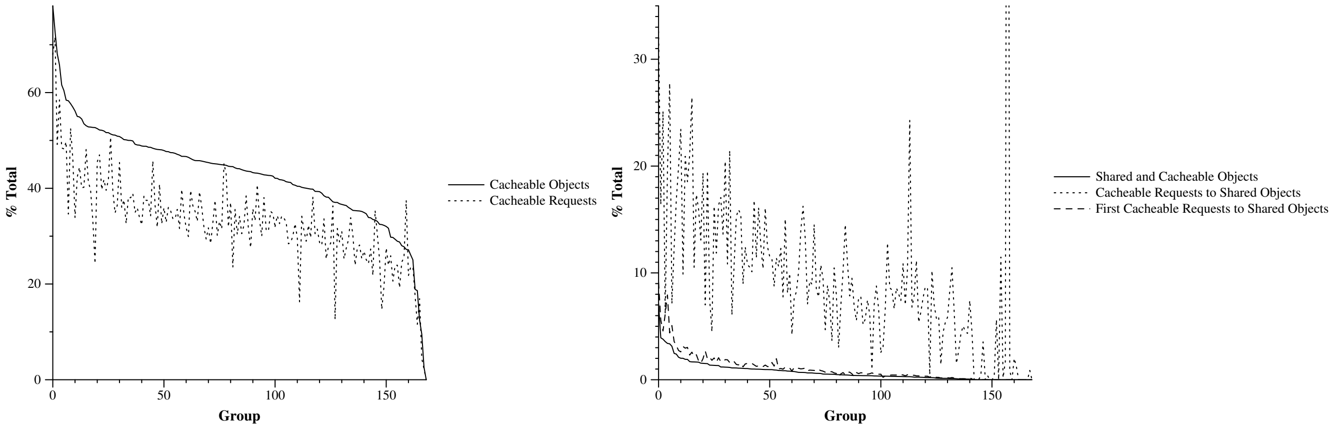
A last factor that can affect the performance of caching is object expiration time. We found overall that only 9.3% of requests had an expiration specified. Most of these requests are to objects that expire quickly; 50% are to objects that expire in less than 2 hours. Interestingly, of those that did have an expiration specified, 43% had a missing or invalid date and 18% had an expiration time that had already passed.

Finally, we have not presented detailed cache simulations here; our objective is simply to analyze cacheability of documents in the most recent data. From our data, it is clear that the trends with respect to cacheability of documents are getting worse. Without special mechanisms to deal with caching, the benefits of proxy caching are likely to deteriorate.

7 Conclusions

In this paper, we have collected and analyzed a large recent trace taken in a university environment. Our study has focused on sharing of web documents within and among a diverse set of organizations within a large university.

We can reach the following conclusions from our data:



(a) Cacheable Objects and Requests

(b) Cacheable and Shared

Figure 9: Fraction of objects and requests accessed by the organization that are potentially cacheable. Requests are not as cacheable as objects due to the reasons described above.

- Organization membership is statistically significant: members of an organization are more likely to request the same documents than a set of clients of the same size chosen at random from all the clients in the population. However, the vast majority of the requests made (and the objects requested) are to objects that are shared among multiple organizations.
- Objects that are simultaneously shared locally by an organization and globally with other organizations are more likely to be requested by an organization member than objects that are just shared locally or just shared globally. This suggests that the most-requested objects by an organization are globally and universally popular.
- The trace shows mostly minor differences relative to earlier traces in terms of most of the basic characteristics. We find larger differences in average object size, number of embedded documents and retrieval latencies. The most significant difference we see relative to previous traces is in client request rate and the appearance of a significant audio/video component.

We have only begun to analyze the data we have collected. Future work includes a much more detailed statistical analysis of various aspects of the data already collected as well as a study of the evolution of WWW traffic characteristics over time. Towards this end, we plan to repeatedly trace and examine Web traffic at the University of Washington.

References

[1] Jussara Almeida, Virgilio Almeida, and David Yates. Measuring the behavior of a World Wide Web server. Technical Report 96-025, Boston University, October 1996.

[2] Virgilio Almeida, Azer Bestavros, Mark Crovella, and Adriana deOliveira. Characterizing reference locality in the WWW. Technical Report 96-011, Boston University, June 1996.

[3] Martin F. Arlitt and Carey L. Williamson. Web server workload characterization: The search for invariants. In *Proc. of the ACM SIGMETRICS '96 Conference*, April 1996.

[4] Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *Proceedings of IEEE INFOCOM '99*, March 1999.

[5] Ramon Caceres, Fred Douglis, Anja Feldmann, Gideon Glass, and Michael Rabinovich. Web proxy caching: the devil is in the details, June 1998.

[6] Pei Cao. Characterization of web proxy traffic and Wisconsin proxy benchmark 2.0, November 1998.

[7] Pei Cao, Jin Zhang, and Kevin Beach. Active cache: Caching dynamic contents on the web. In *Proc. of IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing (Middleware '98)*, September 1998.

[8] Anawat Chankhunthod, Peter B. Danzig, Chuck Neerdaels, Michael F. Schwartz, and Kurt J. Worrell. A hierarchical Internet object cache. In *Proc. of the 1996 USENIX Technical Conference*, January 1996.

[9] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and

possible causes. In *Proc. of the ACM SIGMETRICS '96 Conference*, April 1996.

[10] Carlos R. Cunha, Azer Bestavros, and Mark E. Crovella. Characteristics of WWW client-based traces. Technical Report BU-CS-95-010, Boston University, July 1995.

[11] Fred Douglis, Anja Feldmann, Balachander Krishnamurthy, and Jeffrey Mogul. Rate of change and other metrics: a live study of the World Wide Web. In *Proc. of the USENIX Symposium on Internet Technologies and Systems*, November 1997.

[12] Brian Duska, David Marwood, and Michael J. Feeley. The measured access characteristics of World Wide Web client proxy caches. In *Proc. of the USENIX Symposium on Internet Technologies and Systems*, November 1997.

[13] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z. Broder. Summary cache: A scalable wide-area web cache sharing protocol. In *Proceedings of ACM SIGCOMM '98*, August 1998.

[14] Anja Feldmann, Ramon Caceres, Fred Douglis, Gideon Glass, and Michael Rabinovich. Performance of web proxy caching in heterogeneous bandwidth environments. In *Proceedings of IEEE INFOCOM '99*, March 1999.

[15] Steven D. Gribble and Eric A. Brewer. System design issues for Internet middleware services: Deductions from a large client trace. In *Proc. of the USENIX Symposium on Internet Technologies and Systems*, November 1997.

[16] James Gwertzman and Margo Seltzer. The case for geographical push caching. In *Proc. of the Fifth Annual Workshop on Hot Operating Systems*, May 1995.

[17] Thomas M. Kroeger, Darrell D. E. Long, and Jeffrey C. Mogul. Exploring the bounds of web latency reduction from caching and prefetching. In *Proc. of the USENIX Symposium on Internet Technologies and Systems*, November 1997.

[18] Michal Kurcewicz, Wojtek Sylwestrzak, and Adam Wierzbicki. A distributed WWW cache. In *3rd International WWW Caching Workshop*, June 1998.

[19] Bruce A. Mah. An empirical model of HTTP network traffic. In *Proceedings of IEEE INFOCOM '97*, April 1997.

[20] Steve McCanne and Van Jacobson. The BSD Packet Filter: A new architecture for user-level packet capture. In *Proc. of the USENIX Technical Conference*, Winter 1993.

[21] Jeffrey C. Mogul. Network behavior of a busy web server and its clients. Technical Report 95/5, Digital Equipment Corporation Western Research Laboratory, October 1995.

[22] Michael Rabinovich, Jeff Chase, and Syam Gadde. Not all hits are created equal: Cooperative proxy caching over a wide area network. In *3rd International WWW Caching Workshop*, June 1998.

[23] Squid internet object cache, http://squid.nlanr.net.

[24] Lixia Zhang, Sally Floyd, and Van Jacobson. Adaptive web caching. In *Proc. of the 1997 NLANR Web Cache Workshop*, June 1997.