Changes in the Web from 2000 to 2007

Ramin Sadre, Boudewijn R. Haverkort

University of Twente
Centre for Telematics and Information Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
P.O. Box 217, 7500 AE Enschede, The Netherlands
{r.sadre, b.r.h.m.haverkort}@utwente.nl

Abstract. The World Wide Web has undergone major changes in recent years. The idea to see the Web as a platform for services instead of a one-way source of information has come along with a number of new applications, such as photo and video sharing portals and Wikis.
In this paper, we study how these changes affect the nature of the data distributed over the World Wide Web. To do so, we compare two data traces collected at the web proxy server of the RWTH Aachen. The first trace was recorded in 2000, the other more than seven years later in 2007. We show the major differences, and the similarities, between the two traces and compare our observations with other work. The results indicate that traditional proxy caching is no longer effective in typical university networks.

1 Introduction

Originally designed as a transport protocol for hypertexts, HTTP has become the leading "container" protocol for a large variety of applications in the last decade. Where in the past the web was primarily used for exchanging text-based information, possibly with some pictures, the web now is a major source of both texts and pictures of all sorts, but also of videos and music files. Yet, HTTP is being used for downloading most of these.

In 2000 we collected a trace from the RWTH proxy server, primarily to study object size distributions. Now, seven years later, we collected a similar trace, at the same "point" in the internet, and compared it, in many ways, with the 2000 trace. Apart from the object size distribution, we now also studied object types, and object constellation (sub-objects, links, and so on). In doing so, our aim has been to understand changes in the network traffic (volume as well as other characteristics) as generated by a large population of world-wide web users and how those changes affect the efficiency of the proxy cache.

We are not the first ones to study the characteristics of world-wide web traffic. Many studies have been reported so far (see the references at the end of this paper), however, not many do a comparison in time, as we do. Of the true comparable studies we found, we mention three. In [1], the authors study traces collected in 1999, 2001, and 2003 at the up-link of the University of North Carolina. Since they use TCP/IP headers only they do not examine the type of

the documents requested by the clients. In [2], the accesses to three universitary servers are studied; instead we analyze the accesses of clients located in a university network to the whole WWW. Finally, [3] compares two traces collected at the Boston University in 1995 and 1998. It mainly focuses on the impact of the changes in the traces on different caching algorithms.

The remainder of the paper is structured as follows. In Section 2 we present the general characteristics of the two traces. In Section 3 we make a detailed comparison of various aspects of the traces: the response size and the type of the transferred documents (Section 3.1), the characteristics of the queried URLs (Section 3.2), the structure of the web pages (Section 3.3), and the cache efficiency of the proxy server (Section 3.4). We compare our major observations and conclusions with other work in Section 4. Finally, Section 5 summarizes the paper.

2 General characteristics

The two traces that we examine in this paper have been extracted from the access log files of the Squid web proxy server [4] of the RWTH Aachen. The access log files have been collected at the proxy server from February 17, 2000 to March 12, 2000, respectively from August 6, 2007 to September 4, 2007. Both time periods are similar in the sense that they both fall into the semester breaks of the university. In 2000, main users of the university network were around 1950 scientists employed at the university and around 4350 students (out of 27400) living in student apartment blocks. These numbers have not changed much from 2000 to 2007.

For the following studies we have focused on the HTTP-GET requests that were processed with HTTP status code 200 (OK) or 304 (Not Modified) which were by far the most frequent status codes in the log files. In the following we denote the resulting data sets the "2000 trace", respectively the "2007 trace". We used parts of the 2000 trace in previous publications [5–9].

	2000 trace	2007 trace
start date	2000-02-17	2007-08-06
end date	2000-03-12	2007-09-04
#requests	26,318,853	18,747,109
#requests 200	20,734,319	13,621,849
#requests 304	5,584,534	5,125,260
#clients	2831	1124
#requests\ICP	19,716,054	16,415,251
#clients\ICP	2787	1119
#reqs/client\ICP	7074	14670
volume [Gbytes]	237	866

Table 1. General characteristics

Table 1 shows some general characteristics of the two traces. Row 4 gives the overall number of requests recorded in the traces. Row 5 and 6 give the number of requests by status code. The (relatively) increased number of requests with return code 304 in the year 2007 indicates that client-side caching is now more common and efficient. Row 7 gives the number of unique clients that sent requests to the proxy server. These rows show that less queries have been sent by less clients to the server in the year 2007 than in the year 2000. It should be noted that not all clients represent single hosts since the proxy server is also queried by other proxy servers via the Internet Cache Protocol (ICP) [10]. When ignoring those other proxy servers, we obtain the numbers shown in rows 8 and 9. We observe that the number of clients decreased while the average number of requests per client (row 10) has increased by about 100% from 2000 to 2007. An explanation for the latter will be discussed in Section 3.3. The exact reason for the smaller number of clients is difficult to find because each department and student apartment block of the university is independently administrated and, hence, no information on the employed browser configurations is available.

Whereas the total number of requests has decreased, much more bandwidth has been consumed by responses with status code 200 in 2007 than in 2000. Row 11 shows the volume of transferred documents measured in Gbytes. The cause of this increase will be discussed in Section 3.1.

3 Detailed comparison

3.1 Response size and type

In this section we study the size and type distribution of the responses with status code 200, i.e., responses that transferred an entire document to the client.

	2000 trace	2007 trace
min	17	85
max	$0.228 \cdot 10^9$	$2.147 \cdot 10^9$
mean	12294.0	68275.2
median	2410	2780
SCV	320.9	3425.1

Table 2. Response size statistics (in bytes)

Response size distribution Table 2 gives some important statistics of the response sizes (in bytes) as observed in the two traces. Although the median is similar for both traces, there are extraordinary differences between the two traces concerning the mean and the squared coefficient of variation of the response size distribution (SCV). In previous work [5, 7, 9], we observed that the response size distribution in the 2000 trace is heavy-tailed. The results shown

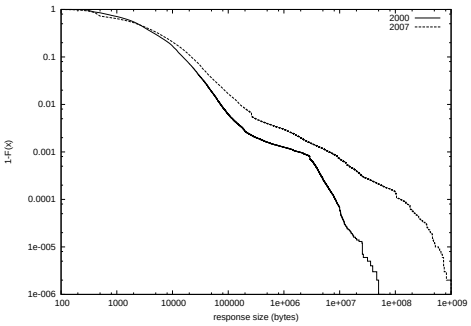

Fig. 1. CCDF of the response size

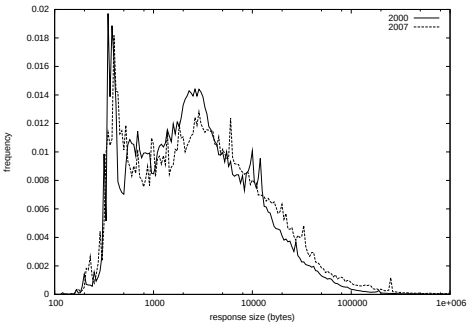

Fig. 2. Response size frequency (logarithmic bin size and normalized frequency)

in Table 2 suggest that the degree of heavy-tailedness has further increased over the last years. This is demonstrated by the log-log plots of the complementary cumulative distribution function (CCDF) of the response size shown in Figure 1. We note that the CCDF of the 2007 trace decays significantly slower for very large response sizes than the CCDF of the 2000 trace, i.e., it is more heavy-tailed.

We are also interested in the changes of the distribution of smaller response sizes. Figure 2 shows for the two traces the histogram of the response sizes with logarithmic bin sizes and frequencies normalized to the trace size. As already expected from the increased degree of heavy-tailedness, we observe that the histogram of the more recent trace has a less developed waist in the range of 2000–5000 bytes, whereas the number of responses larger than 10000 bytes has considerably increased over a wide range. In addition, we observe that both histograms exhibit several distinct peaks. Explanations for these peaks can be found by a deeper analysis of the transferred documents in the following section.

Object types To determine the types of the transferred documents we use the MIME type [11] information found in the Squid log files. Although the sent MIME type can be freely chosen by the server it is the only source of information for objects dynamically generated by server-side scripts or applications. We found 376 different MIME types in the 2000 trace and 384 different types in the 2007 trace. Table 3 show the frequencies of the ten most frequent types in the 2000 trace, respectively the 2007 trace. We observe a slight diversification of the types: In 2007, the five most frequent types only cover 88.2% of all requests, compared to 98.1% in 2000. Furthermore, it shows that the JPEG format and the PNG format have gained popularity over the GIF format. This could, however, be a consequence of improved client-side caching since GIF images are, in general, very small objects as shown in the following.

A different ranking is obtained when we examine the bandwidth consumed by the different types. Table 4 shows the percentage of the overall traffic volume (second column) consumed by the ten most bandwidth-consuming types in the

type	#
image/gif	53.2%
image/jpeg	24.9%
text/html	18.4%
application/x-javascript	1.1%
text/plain	0.5%
text/css	0.5%
application/octet-stream	0.3%
image/pjpeg	0.2%
(unspecified)	0.1%
video/mpeg	0.1%

type	#
image/jpeg	33.3%
image/gif	28.5%
text/html	16.0%
application/x-javascript	6.9%
image/png	3.5%
text/css	2.5%
text/plain	1.8%
text/javascript	1.7%
application/octet-stream	1.3%
application/x-shockwave-flash	1.2%

Table 3. Number of requests by type for the 2000 trace (left) and the 2007 trace (right) as percentage of the trace size.

type	volume	size	cat
image/jpeg	21.5%	10	i
image/gif	15.5%	4	i
text/html	14.6%	9	t
application/msword	9.0%	4147	f
application/octet-stream	8.4%	672	f
application/zip	8.1%	1322	f
video/mpeg	6.8%	861	v
application/vnd.ms-excel	2.5%	3637	f
text/plain	2.2%	49	t
audio/mpeg	2.1%	3360	a

type	volume	size	cat
application/octet-s.[1]	34.6%	1766	v/f
image/jpeg	6.6%	13	i
application/x-otrkey	6.6%	240610	f
text/plain	6.1%	231	t
video/x-msvideo	6.0%	109533	v
video/x-flv	5.9%	10954	v
video/flv	5.4%	6730	v
video/x-ms-wmv	3.2%	42636	v
text/html	3.1%	13	t
application/zip	2.5%	9632	f

Table 4. Fraction of traffic volume and average response size (in Kbytes) by type for the 2000 trace (left) and the 2007 trace (right) (i: image; t: text; v: video; a: audio; f: formatted). [1] = application/octet-stream

2000 trace, respectively the 2007 trace, and the corresponding average response size for documents of these types (third column). The last column gives a rough categorization of the content type. While in the year 2000, most of the traffic volume was caused by "traditional" web site components (i.e., HTML documents and images), we see that in the year 2007 nearly all traffic volume is generated by either video clips or archive-file downloads.[1] The average response sizes illustrate that documents of these types are responsible for the higher mean response size of the 2007 stream, as observed in Section 3.1. However, even the "traditional" types HTML and JPEG have increased in size by about 30%.

The response size distributions for the different document types help to understand the shape of the size distribution of the whole trace (see Section 3.1). In Figure 3, we show for both traces the response size histogram (with logarithmic bin size) for the whole trace (labeled "all'), for the most frequent document

[1] Note that the MIME type `application/octet-stream` is not very meaningful here. Only an inspection of the file name endings in the traces showed that most of the objects of that type are either video clips or software archives.

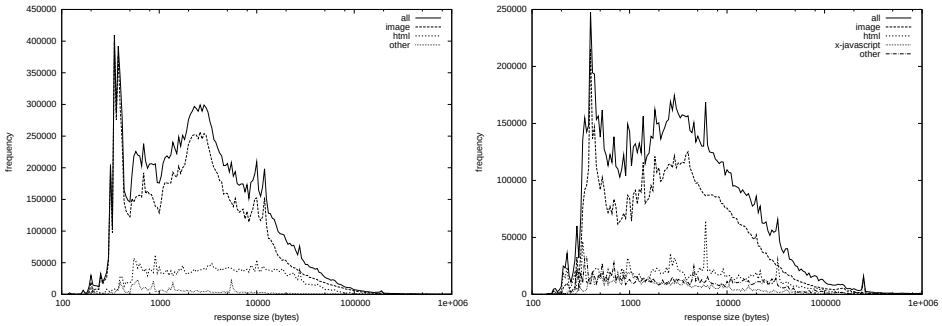

Fig. 3. Response size frequency by object type for the 2000 trace (left) and the 2007 trace (right) (with logarithmic bin size)

types ("images", "html" and, only for the 2007 trace, "x-javascript") and for all remaining types ("other"). Note that we have subsumed all image types (GIF, JPEG, PNG, etc.) under the "image" histogram. We do the following observations:

1. Image objects dominate the shape of the size distribution of both traces.
2. In the 2000 trace, the peaks in the range 300–400 bytes are caused by images from some few, but, at the time of the measurement, very popular web sites. 21% of all image requests in this range refer to only five servers (two of them are banner advertisment servers). In contrast, for the whole trace, the five most frequently queried servers are only accountable for 4.3% of all image requests.
3. In the 2007 trace, the three most distinct peaks are mainly caused by three sites. The peak at byte size 400 in the image histogram is created by about 100000 queries to `www.google-analytics.com`. In the peak at size 6200 in the HTML histogram, about 43000 queries refer to the download page of a file-sharing server. The Google search pages contribute with about 4000 queries to that peak. Similarly, the peak at the right end (byte size 250000) of the histogram nearly entirely consists of binary file downloads related to an online role-playing game.

3.2 Queried URLs

Some important characteristics of the queried URLs are summarized in Table 5. The second row gives the number of unique URLs queried by all clients. Row 3 gives the average number of requests per unique URL. Although the average did not change significantly from 2000 to 2007, the distribution of the number of requests per unique URL did, as illustrated by Figure 4. It shows, for both traces, the number of queries per URL normalized to the total number of queries. The shape of the log-log plot still follows Zipf's law [12, 3, 13], but the slope has changed. As can be seen, the most popular URLs in the 2007 trace are much

	2000	2007
#URLs	6,846,724	4,896,006
#request/#URL	3.84	3.83
#URLs (client)	5282	6038
1x-URLs	65.1%	76.7%
#servers	137832	99147
#URLs/server	49.67	49.38

Table 5. Characteristics of the queried URLs

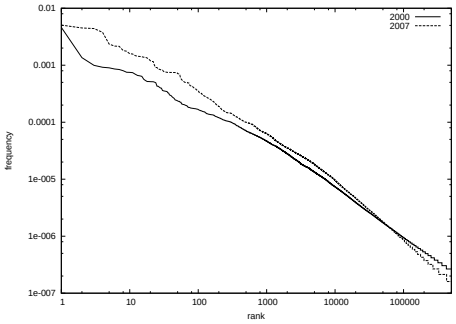

Fig. 4. Normalized number of queries per URL sorted by URL rank

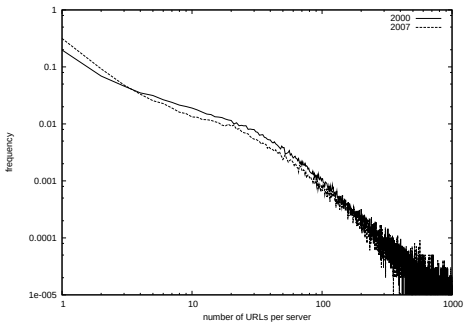

Fig. 5. Histogram of the number of URLs per server normalized to the number of servers

more often requested (in relation to the trace size) than their counterparts in the older trace. Only the most popular URL has a comparable request frequency in both traces. In addition, the average number of distinct URLs a client accesses has increased (row 4). Consequently, we observe an increase of URLs that have been requested only once during the whole measurement period. Row 5 shows their percentage in the two traces.

The last two rows show the number of queried servers (row 6) and the average number of unique URLs per server (row 7). Again, the average did not change while the distribution did as shown in Figure 5 which depicts for both traces the histogram of the number of URLs per server, normalized to the number of servers. We observe that the number of servers with 5 to 100 URLs has decreased. In contrast, the number of servers with only one or two distinct URLs has considerably increased. It is difficult to identify the reason for this change due the limited information stored in the traces. However, a manual inspection of the involved URLs showed that many of the "single-URL" servers are not "real" web sites but servers providing services to other web sites, such as visit counters, advertisements, etc. This implies that the interconnectivity of servers has increased: web servers offer less URLs but more often refer to other servers.

The servers with the largest number of distinct URLs are `www.geocities.com` in the year 2000 (98168 URLs) and `img.youtube.com` in the year 2007 (106675

server	queries
www.spiegel.de	1.02%
informer2.comdirect.de	0.98%
www.geocities.com	0.87%
www.heise.de	0.80%
www.africam.com	0.77%
ad.de.doubleclick.net	0.74%
www.ebay.de	0.68%
www.eplus.de	0.61%
www.consors.de	0.47%
ad.doubleclick.net	0.38%

server	queries
www.spiegel.de	2.03%
image.chosun.com	1.93%
www.svd.se	1.71%
img.youtube.com	0.85%
images.gmx.net	0.76%
www.bild.t-online.de	0.75%
pagead2.googlesyndication.com	0.68%
www.heise.de	0.67%
www.google-analytics.com	0.64%
www.manager.co.th	0.62%

Table 6. Number of queries (as fraction of all queries) by server for the 2000 trace (left) and the 2007 trace (right)

URLs). However, these two servers are not the targets of the largest number of requests, as illustrated by the lists of the ten most queried servers shown in Table 6. We observe that in the year 2000 the five most queried servers received approximately the same number of requests, whereas in the year 2007 the difference between the servers at rank 1 and 5 was much larger.

3.3 Page structure

An important question for the modeling of WWW traffic concerns the structure of web pages, that is, how many documents does a web browser have to fetch from the server in order to download the complete web page (called the *page size* in the following). This question is difficult to answer if proxy server traces are used as the only source of information. Given a sequence of queries sent by a specific client, the hardest problem is to identify the first and the last request for a specific web page.

A popular approach is based on the timestamps of the requests [1, 14–16]. Let a and b be two consecutive requests sent by the same client at time s_a, respectively s_b, with $s_a < s_b$. The responses are sent back to the client at times e_a, respectively e_b. We define that two requests belong to the same web page if $s_b - e_a < \delta_{th}$ where δ_{th} is a predefined threshold. Note that $s_b - e_a$ can be negative if the client software sends requests in parallel. The best value for the threshold is usually unknown. It has to be larger than the time that the client software needs to process a response before it can send the next request. If the the threshold is too large, two requests belonging to two different web pages may be wrongly associated. In [1, 14, 15], a threshold of 1 second is used, assuming that the user certainly needs more than 1 second to react before calling the next web page.

We have calculated for both traces the mean page sizes that result from different thresholds (ignoring requests from other proxy servers, as described in Section 2). The results are shown in Figure 6. The figure illustrates that it is rather problematic to choose a specific threshold, such as 1 second, since the cal-

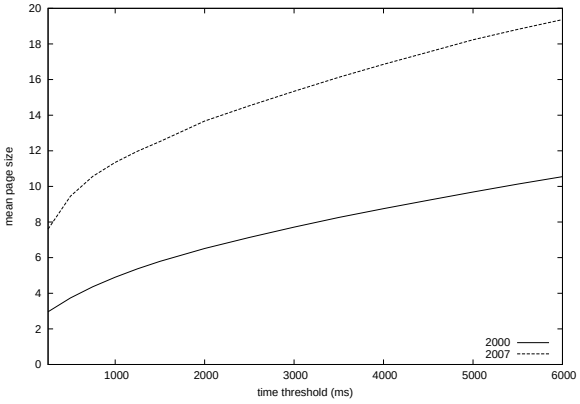

Fig. 6. Mean page sizes determined for different time thresholds for the 2000 and 2007 trace

trace	hit	miss	size hit	size miss
2000	54.3%	45.7%	8980	16230
2007	23.0%	77.0%	9846	85737

Table 7. Cache efficiency and mean response size (in bytes)

trace	total	hit	miss
2000	21.2%	1.2%	20.2%
2007	37.1%	4.3%	32.8%

Table 8. Fraction of dynamically created responses by caching status

culated mean page size continuously increases with the threshold. Nevertheless, we observe for a wide range of thresholds that the page size has doubled from the year 2000 to 2007, which explains very well the 100% raise of the number of requests per client as observed in Section 2. However, we advise to interpret these results with caution for the following reasons:

- Client-side cached documents may not be recorded by the traces.
- The user may call two web pages at the same time.
- Some requests in the trace do not necessarily reflect "normal" human behavior. For example, we have identified a series of 2300 requests in the 2000 trace with very small inter-request times. A user was obviously using a tool to download a complete copy of a web page. In another example, an advertisement banner caused a client to send over 10000 requests.

3.4 Cache efficiency

Statistics about the caching behavior of the proxy server in 2000 and 2007 are shown in Table 7. The column titled "hit" and "miss" show the number of hits, respectively misses, as fraction of the total number of requests in the traces. The columns "size hit" and "size miss" give the mean size of the responses (in bytes) in case of a hit, respectively miss.

We observe a drop of the cache efficiency from 2000 to 2007. Whereas the mean size of cached documents has not changed very much, the probability for

a cache hit significantly decreased. These numbers can not be directly compared because the proxy server has changed in terms of hardware and software from 2000 to 2007. However, we identify several reasons for the large number of cache misses that are more or less independent of the used server configuration:

1. The increased number of *very* large documents, as shown in Section 3.1. Since proxies only have limited cache memory such documents generate cache misses.
2. The increased number of URLs that have been only queried once, as discussed in Section 3.2.
3. The increased number of responses with status code 304 (see Section 2), indicating that client-side caching has improved its efficiency.
4. The increased number of dynamically generated responses, although that number can not be exactly determined because the traces do not include enough information. Especially the HTTP header lines controlling the cache behavior and the expiry dates would be helpful here.

An approach to identify dynamic documents is to scan the trace for URLs that frequently change their response size. This method, however, is not applicable here due to the large number of one-timer URLs. To get a very rough idea about the number of dynamic documents we have analyzed the URLs themselves. We have regarded an URL as dynamic when it contained form data or referred to scripts or server pages (file endings .php, .asp, etc.). Table 8 shows for both traces the resulting number of requests referring to dynamic documents as percentage of the number of all requests in the trace (column 2), of all requests causing a cache hit (column 3) and of all requests causing a cache miss (column 4). It shows that the number of requests to dynamic documents has increased and strongly contributes to the number of cache misses.

4 Comparison to other work

In the following, we give a summary of the important observations presented in the previous sections. For each observation, we give, if available, references to other publications that do (not) support it.

- Responses with status code 304 are more prevalent in 2007 than in 2000 (also reported in [2]).
- The volume of transferred documents (measured in bytes) significantly increased (also observed in [1–3]).
- The heavy-tailedness of the response size distribution increased from 2000 to 2007. The increase has been also reported by [1–3] but not at that extend. For example, the largest document observed in [2] for a trace from the year 2004 had a size of 193 Mbytes, whereas the largest file in our 2007 trace has a size of around 2 Gbytes. The fact that sizes are heavy-tailed distributed has been reported in many previous publications [17–19].

- "Traditional" web site components such as HTML documents and images are still the most frequently transferred objects. However, the most bandwidth consuming documents are now videos and compressed software archives. Other authors have observed similar trends [2] but only for single web servers. We are not aware of recent studies on *proxy* servers of which the workloads, in our opinion, better reflect the overall trends in the Internet. The document type distributions found in the 2000 trace are consistent with the results reported in an older study on proxy server workloads [20].
- We have shown that a few popular servers can have a considerable impact on the size and type distribution of the transferred documents. The phenomenon of *concentration* [17, 21], i.e., the fact that a large part of all requests concentrates on a few popular servers, has significantly increased (also reported by [2]). However, the popularity of the URLs still follows Zipf's law [2, 3, 12, 13, 19].
- The number of servers only hosting one or two URLs has considerably increased. Those servers are mostly providing services to other servers. This implies an increased inter-connectivity of web servers in the Internet, as also observed in [1].
- The number of one-timers (URLs only called once) has increased by 10%. In contrast, [2] reports a decrease by 50% for web servers. This illustrates the diversity of the documents as perceived by a proxy server in contrast to single web servers.
- The complexity of web pages (measured in number of URLs) has increased (confirmed by [1]). Older measurements can be found in [14, 16].
- Cache efficiency for transferred documents has decreased. We believe that one reason is the significant amount of responses that are dynamically generated. Other publications report much lower amounts of dynamic documents [2, 20].
- Our results indicate that traditional proxy servers that aim to cache *all* types of contents are not effective anymore in the studied network (and in networks with similar usage characteristics). However, specialized caching and distribution servers could provide better results as shown by simulation for YouTube videos in [22].

In this paper we have mostly focused on the stationary properties of the Web traffic. Some other publications have studied its time dynamics, especially correlations in the request rate [2], appearance of new documents [23], and the inter-reference time and the locality of references [2, 24, 25].

5 Summary

In this paper we have compared two data traces collected at the web proxy server of the RWTH Aachen. One was collected in 2000, the other seven years later in 2007. Through an elaborate data analysis, we show that the changes the World Wide Web has undergone in recent years have had a major impact on the volume and the nature of the observed traffic. Foremost, the size of the

transferred documents has significantly increased due to the emerging popularity of bandwidth-consuming formats, such as videos and, surprisingly, binaries (software updates). At the same time, web pages have become more complex, i.e., they now consist of more objects, refer more often to other web servers, and more often rely on dynamically generated documents. For these (and some other) reasons, the efficiency of proxy servers has significantly decreased, indicating that traditional, non-specialized proxy servers that aim to cache all types of contents are not effective anymore. On the other hand, our measurement data also implies that client-side caching is now more common and efficient.

Our observations and their comparison to other work indicate that the changes that we have observed in our traces are, to some extend, representative for the global evolution of the World Wide Web. However, we realize that our conclusions are limited by the fact that only one server has been studied in this paper. Hence, we suggest that other researchers do similar comparisons with their old traffic measurements.

As for the future, we plan to continue our study of web traffic in order to observe changing workload patterns. At the same time, the traces we collected can be used for some other interesting studies, such as changes in user behavior and hyper-link topology.

Acknowledgments We thank Wilfred Gasper, Rechenzentrum RWTH Aachen, for providing the anonymized 2007 trace.

References

1. Hernandez-Campos, F., Jeffay, K., Smith, F.: Tracking the Evolution of Web Traffic: 1995-2003. 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003 (MASCOTS 2003) (2003) 16–25
2. Williams, A., Arlitt, M., Williamson, C., Barker, K.: Web workload characterization: Ten years later. In: Web Content Delivery. Springer (2005) 3–21
3. Barford, P., Bestavros, A., Bradley, A., Crovella, M.: Changes in Web client access patterns: Characteristics and caching implications. World Wide Web **2**(1-2) (1999) 15–28
4. Squid: `www.squid-cache.org` (2007)
5. El Abdouni Khayari, R., Sadre, R., Haverkort, B.: Fitting World-Wide Web request traces with the EM-algorithm. In: Proc. of SPIE 4523 (Internet Performance and Control of Network Systems). (2001) 211–220
6. El Abdouni Khayari, R., Sadre, R., Haverkort, B., Zoschke, N.: Weighted fair queueing scheduling for World Wide Web proxy servers. In: Proc. of SPIE 4865 (Internet Performance and Control of Network Systems III). (2002) 120–131
7. El Abdouni Khayari, R., Sadre, R., Haverkort, B.: Fitting World-Wide Web request traces with the EM-algorithm. Performance Evaluation **52**(2-3) (2003) 175–191
8. Haverkort, B., El Abdouni Khayari, R., Sadre, R.: A class-based least-recently used caching algorithm for World-Wide Web proxies. In: Computer Performance Evaluations, Modelling Techniques and Tools. 13th International Conference, TOOLS 2003. Volume 2794 of Lecture Notes in Computer Science., Springer (2003) 273–290

9. Sadre, R., Haverkort, B.: Fitting Heavy-Tailed HTTP Traces with the New Stratified EM-algorithm. In: IT-NEWS 2008 - 4th International Telecommunication NEtworking Workshop on QoS Multiservice IP Networks, 2008 (QoS-IP). (2008) 254–261
10. Network Working Group: RFC 2186. Internet Cache Protocol (ICP), version 2. `http://tools.ietf.org/html/rfc2186` (1997)
11. Network Working Group: RFC 2045. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. `http://tools.ietf.org/html/rfc2045` (1996)
12. Zipf, G.: Relative frequency as a determinant of phonetic change. Reprinted from the Harvard Studies in Classical Philiology. Volume XL. Linguistic Society of America (1929)
13. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.: Web Caching and Zipf-like Distributions: Evidence and Implications. In: INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Computer Society (1999) 126–134
14. Mah, B.: An Empirical Model of HTTP Network Traffic. In: INFOCOM '97: Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE Computer Society (1997) 592–600
15. Barford, P., Crovella, M.: Generating representative web workloads for network and server performance evaluation. In: SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems. (1998) 151–160
16. Arlitt, M.: Characterizing web user sessions. ACM SIGMETRICS Performance Evaluation Review **28**(2) (2000) 50–63
17. Arlitt, M., Williamson, C.: Internet Web Servers: Workload Characterization and Performance Implications. IEEE/ACM Transactions on Networking **5**(5) (1997) 631–645
18. Crovella, M., Bestavros, A.: Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. IEEE/ACM Transactions on Networking **5**(6) (1997) 835–846
19. Crovella, M.: Performance Characteristics of the World Wide Web. In: Performance Evaluation: Origins and Directions, Springer-Verlag (2000) 219–232
20. Mahanti, A., Williamson, C., Eager, D.: Traffic analysis of a web proxy caching hierarchy. EEE Network, Special Issue on Web Performance **14**(3) (May/June 2000) 16–23
21. Arlitt, M., Jin, T.: A workload characterization study of the 1998 World Cup Web site. IEEE Network **14**(3) (2000) 30–37
22. Zink, M., Suh, K., Gu, Y., Kurose, J.: Watch global, cache local: Youtube network traffic at a campus network: measurements and implications. Multimedia Computing and Networking 2008 **6818**(1) (2008) 681805–681817
23. Cherkasova, L., Karlsson, M.: Dynamics and Evolution of Web Sites: Analysis, Metrics and Design Issues. In: Sixth IEEE Symposium on Computers and Communications (ISCC '01), IEEE Computer Society (2001) 64–71
24. Almeida, V., Bestavros, A., Crovella, M., de Oliveira, A.: Characterizing reference locality in the WWW. In: Fourth international conference on on Parallel and distributed information systems (DIS '96), IEEE Computer Society (1996) 92–107
25. Arlitt, M., Friedrich, R., Jin, T.: Performance Evaluation of Web Proxy Cache Replacement Policies. In: TOOLS '98: Proceedings of the 10th International Conference on Computer Performance Evaluation: Modelling Techniques and Tools, Springer-Verlag (1998) 193–206