Web Proxy Workload Characterization

Anirban Mahanti
Carey Williamson
Department of Computer Science
University of Saskatchewan

February 17, 1999

Abstract

This document presents a workload characterization study for Web proxy servers. Three different Web proxies are studied: the Web proxy cache at the University of Saskatchewan; the primary CANARIE Web proxy cache; and an NLANR Web proxy cache at the University of Illinois, Urbana-Champaign. Workload characterization is done using access log analysis from the three sites, with access log durations ranging from 1 to 3 months.

Our study identifies several workload characteristics of interest. For example, HTML and image documents together account for over 95% of the requests and documents seen in the workloads studied; the document size and transfer size distributions are heavy-tailed; the reference frequency per document is Zipf-like in its distribution, but not precisely a Zipf distribution; among all the documents accessed in the proxy logs, approximately 70% are accessed only once in the log; the top 30% of the documents account for 80% of the requests; the "hot set" of active documents changes relatively slowly on a day-to-day basis during the week, but quite significantly on weekends; and typical hit rates achieved by proxy caches in a Web caching hierarchy range from 10-40%. We find these characteristics quite consistent across the three different levels of caching hierarchies studied in the traces, though the workload on the CANARIE Web proxy cache is much lighter than those for the other two sites.

1 Introduction

This document presents a workload characterization study for Web proxy servers, as part of a larger project on the design and performance evaluation of a national-scale Web caching hierarchy for CA*net II.

The purpose of the workload characterization study is two-fold. First, the study provides a snapshot reflecting the operation of the CA*net II Web caching hierarchy in its current form (e.g., volume of workload, cache hit rates, document transfer times). Second, the study identifies common characteristics in the workloads presented to Web proxy servers. Understanding these characteristics is an important step in the process of improving the caching hierarchy, thereby reducing network traffic and providing documents to users with reduced latencies.

In this study, we characterise Web proxy workloads at both the institutional level (e.g., regional or institutional Web proxy cache) and at a higher level (e.g., national and/or international Web proxy cache). The workload characterization is carried out using Web proxy access log analysis, similar to the Web server access log analysis carried out by Arlitt and Williamson [4]. Throughout the study, emphasis is placed on identifying workload characteristics that apply across the levels of a Web caching hierarchy.

The main observations from our workload characterization study are the following:

- HTML and image documents together account for over 95% of the requests and documents seen in the Web proxy access logs.
- The document size and transfer size distributions are heavy-tailed (i.e., Pareto, with $\alpha \approx 1.30$).
- The number of references to each document identified in the access log follows a Zipf-like distribution, but does not precisely fit a Zipf distribution.
- Among all the documents accessed in the proxy logs, approximately 70% of the documents are accessed only once in the log.
- The top 30% of the accessed documents account for over 80% of the requests seen in the access logs.

- The "hot set" of active documents changes relatively slowly on a day-to-day basis during the week, but changes quite significantly during evenings and on weekends.

- Typical cache hit rates for Web proxies in a caching hierarchy are 10-40%.

We find these characteristics quite consistent across the three different levels of caching hierarchies studied in the traces. Our study also shows that the CANARIE Web proxy cache is not very well utilized in the current CA*net II architecture; its workload is much lighter than those for the other two Web proxies studied.

The remainder of this document is organized as follows. Section 2 describes the data collection methodology and the sites used for this study. This is followed by an analysis of the raw data sets in Section 3. Section 4 presents our workload characterization results for Web proxy servers. Finally, Section 5 summarizes our work and presents conclusions.

2 Data Collection Methodology

Our workload characterization study is conducted using analysis of access logs from Web proxy servers. Each entry in the access log records the URL of the document being requested, the date and time of the request, the name (or IP address) of the requesting client, the type of the document, and additional information such as whether the document was found in the proxy cache or not, a response code, and the number of bytes returned to the requesting client. Processing these log entries can produce summary statistics about workload volume, document types and sizes, document popularity, and proxy cache performance.

The access logs for our study were obtained from three World-Wide-Web proxy servers:

- University of Saskatchewan

- CANARIE

- NLANR

Each of these sites continuously records access logs, which we obtained on a daily basis using ftp. The data sets for the three sites currently range from 1 month to 3 months in duration. Further detail on each of these sites[1] is provided below.

The Web proxy server at the University of Saskatchewan represents an institutional-level Web proxy, functioning as a secondary-level (i.e., lower level) proxy cache in the CA*net II caching hierarchy. It serves several hundred users on the University of Saskatchewan campus who have configured their browsers to use the proxy cache. This proxy server is operated by the Department of Computing Services at the University of Saskatchewan. The proxy uses a Digital AlphaServer 1200 5/400 with two 400 MHz processors running Squid version 1.xx [16]. The proxy is configured to use the CANARIE cache as a parent, using the Squid Inter-Cache Protocol (ICP). That is, cache misses at the University of Saskatchewan cache result in requests to the CANARIE cache for the missing document. Requests for dynamic content are configured to bypass the proxy cache entirely.

The CANARIE proxy cache is the primary-level core of the CA*net II caching hierarchy. This machine (called *roadrunner*) is a SPARC Ultra 180 MHz machine, running Squid version 1.2.beta22 [7]. It has 512 MB RAM and a 30 GB hard disk. The CANARIE proxy server is physically located at Bell Canada in Toronto, though it is administratively controlled by the CANARIE ARDNOC (Advanced Research and Development Network Operations Center). This cache currently functions as a parent for several secondary-level proxies, including University of Saskatchewan, University of Alberta, Dalhousie University, and McMaster University. There are also a small number of users who configure their browsers to directly use the CANARIE proxy cache. The CANARIE proxy has parent links to two nodes in the NLANR (National Laboratory for Applied Networking Research) caching hierarchy, namely the Pittsburgh NLANR node, and the NLANR node at the University of Illinois, Urbana-Champaign (UIUC).

The NLANR traces in our study come from the NCSA (National Center for Supercomputing Applications) proxy server at the University of Illinois, Urbana-Champaign (UIUC). This site represents one of several top-level nodes in the NLANR Web caching hierarchy; it receives requests from sib-

[1] We are always looking for access logs from other proxy servers, to obtain an even better understanding of Web proxy servers and their performance. If you know of additional sites with access logs available, please let us know.

ling caches at the top level, as well as from lower-level caches that use it as a parent. The NCSA proxy server is a Digital AlphaServer 1000 266 MHz machine [23] running Squid version 1.2.beta17 [17].

The access logs from the six root caches in the US can be obtained by FTP from `ftp://ircache.nlanr.net/Traces/`. We regularly downloaded the access logs for the Urbana-Champaign cache since the access logs are updated on a weekly basis (note, however, that the sanitized client IP address mappings are changed on a daily basis). The access logs from the CA*net II cache were obtained by ftp from `http://ardnoc41.canet2.net/cache/squid/rawlogs/`. The CA*net II logs were first made available in December 1998, and are updated on a regular basis. The access logs for the Saskatchewan cache were available directly from the operators of the cache.

The three data sets described above will be referred to as USask, CANARIE, and NLANR in the rest of the document. We present the logs in this relative order to reflect the progression from institutional-level Web proxy cache to top-level Web proxy cache.

3 Access Log Analysis

3.1 Raw Data Analysis

The first step towards analysing the access logs was to concatenate the log files of individual days together to obtain longer data sets for each site. We created a trace spanning the months of October through December, 1998 for the University of Saskatchewan proxy server, which is referred to as the USask access log. The three-month USask access log recorded a total of 27,259,778 requests in 82 days of activity.[2] Similarly, a one-month-long trace for the CANARIE proxy and the NLANR proxy were created. The CANARIE access log recorded a total of 27,627,843 requests in 26 days of activity. [3] The NLANR access logs recorded 28,522,256 requests in the 31 days of activity in the month of December, 1998. Table 1 provides a summary of the access logs for all three proxy servers.

The access logs provide information on proxy servers with different workloads. The USask proxy serves users at the University of Saskatchewan,

[2]The access logs for 10 days were not available, due to downtime for server upgrades and network outages.

[3]The access logs for 5 days in December, 1998 were not available.

Table 1: Summary of Web Proxy Access Log Characteristics (Raw Data)

Item	USask	CANARIE	NLANR
Access Log Duration	82 days	26 days	31 days
Start Date	Oct 2, 1998	Dec 1, 1998	Dec 15, 1998
End Date	Dec 31, 1998	Dec 31, 1998	Jan 14, 1999
Total Requests	27,259,778	27,627,843	28,522,256
Avg Requests/Day	332,436	1,062,609	920,072
Total Bytes Transferred (GB)	164	36	301
Avg Bytes/Day (MB)	2003	1406	9695

and therefore its clients are mostly individual users. The CANARIE cache was configured as a parent for the USask cache. Its clients are mostly institutional-level proxies (University of Saskatchewan, University of Alberta, CA*net II Networks Operations Center, Communications Research Center) [7]. The CANARIE proxy is also a child to two NLANR proxies, namely the PSC (Pittsburgh Supercomputing Center) cache at Pittsburgh, Pennsylvania and the NCSA cache at Urbana-Champaign, Illinois. The NCSA cache at Urbana-Champaign mostly serves requests from within the US and mostly consists of clients from North America. Most of the clients for the NCSA cache are institutional caches or other proxy caches like the CANARIE cache.

Table 1 provides insight into the activity of each of these proxy servers. The CANARIE and NLANR caches receive three times more requests per day than the USask cache. Although the CANARIE cache has the highest activity in terms of the number of requests received per day, the average volume of bytes transferred by it was the lowest among the three workloads considered. On further analysis of the CANARIE access log, it was found that approximately 85% of the requests were Squid Inter-Cache Protocol (ICP) queries to the ICP port, which do not result in any transfer of documents (i.e., the access log records a transfer of zero bytes for them). Thus the actual workload (e.g., document transfers) for the CANARIE proxy is much less than that suggested in Table 1. For the USask proxy, 9% of the recorded requests were ICP queries, while for the NLANR proxy, none of the requests were ICP queries, since the NLANR access log was not configured to record activity at the ICP port. These results are summarised in Table 2.

Table 2: Breakdown of Request Methods

Method	USask	CANARIE	NLANR
GET	90.58	15.38	99.94
ICPQUERY	8.68	84.61	0.00
Others	0.74	0.01	0.06
Total	100.0	100.0	100.0

In this study, we are interested in the activity at the HTTP port (i.e., requests for and transfers of Web documents). For this purpose, we study the response codes in the access logs for all TCP requests seen (i.e., activity at the HTTP port). The breakdown of the HTTP reply codes as a percentage of the total number of TCP requests seen is provided in Table 3.

Table 3: Breakdown of HTTP Response Codes

Response Code	USask	CANARIE	NLANR
200 (OK)	83.65	81.00	64.81
206 (Partial Contents)	0.36	0.09	0.27
302 (Found)	5.83	4.36	4.58
304 (Not Modified)	7.35	10.94	26.28
Others	2.80	3.61	4.06
Total	100.00	100.00	100.00

There are many possible responses that a Web proxy can provide to a client request [28]. A reply code of 200 (OK) means that a valid document was made available to the client, either directly from the proxy cache (TCP_HIT), or by retrieving the document from another proxy cache or from the originating server (TCP_MISS). A reply code of 304 (Not Modified) implies that the client had issued a GET If-Modified-Since request to determine whether the client's cached copy of a document was up-to-date or not, and the server (or some intermediate proxy) replied indicating that the client has a valid copy of the document. An HTTP response code of 206 (Partial Contents) implies a partial transfer of the document to the client, while an HTTP response of 302 (Found) means that the requested document is known to reside in a different location than that specified by the URL.

3.2 Data Reduction and Analysis

The access logs record the number of bytes transferred to the clients, regardless of where the requested document was found (e.g., at the proxy cache, at some other cache in the hierarchy, or at the originating server). While characterizing the workload of the proxy servers, we are interested in all requests which would result in the document being accessed from the origin server, had there been no intermediate proxy. In other words, the objective is to understand the effectiveness of proxy caching. For example, suppose that a client issues a GET If-Modified-Since request to the proxy, meaning that the client wants to verify whether or not its cached copy of a document is still valid. In response, the proxy might indicate that the client does *not* have a valid copy of the document, but the proxy can provide a valid copy of the document from its cache [29]. These requests are logged as an HTTP 304 Not Modified response (TCP_REFRESH_HIT) in the access logs. Since this behaviour is purely due to caching at the proxy server, these requests are considered along with all HTTP 200 (OK) responses and HTTP 206 (Partial Contents) responses in the remaining analyses in this paper. We reduce our raw access logs to consider only these cases, which reflect successful transfers of documents to the requesting clients.

Table 4 summarizes the reduced access logs for our three Web proxy sites. Based on the average number of requests seen per day at each proxy server, the NLANR proxy server has the highest activity, while the CANARIE proxy server has the least activity. This is not surprising since very few institutional caches currently use the CANARIE proxy cache.

Table 4 also indicates the number of distinct documents, servers, and clients recorded in the access logs. Each proxy handles requests for millions of distinct documents from thousands of different Web servers. The number of clients (i.e., distinct client IP addresses seen) varies quite significantly in the three traces. This number is not known precisely for the NLANR log, since the client IP addresses in the NLANR access logs are randomized every day (due to privacy concerns). However, on any particular day, about 700 clients generate requests to the NLANR cache.

Overall, the mean and median document transfer sizes are quite small, as has been reported in previous studies of Web servers [3, 4, 8] and Web proxy servers [1]. In our data sets, the mean size of the documents transferred ranges from 7-15 kilobytes, while the median is in the range of 2-3 kilobytes. The mean transfer size is larger than the median transfer size because there are several large documents that skew the mean of the transfer size distribution. There is also high variability in the sizes of documents transferred, as indicated by the coefficient of variation (COV) reported in Table 4.

Table 4: Summary of Web Proxy Access Log Characteristics (Reduced Data)

Item	USask	CANARIE	NLANR
Total Requests	20,754,720	351,296	20,018,680
Avg Requests/Day	253,106	135,103	645,763
Total Bytes Transferred (GB)	160	33	295
Avg Bytes/Day (MB)	1,964	1,284	9,562
Distinct Documents	5,527,667	1,423,081	7,681,214
Distinct Servers	110,685	19,214	211,555
Distinct Clients	???	11	1,200
Mean Transfer Size (Bytes)	7,761	9,505	14,808
Median Transfer Size (Bytes)	2543	2819	3112
Coefficient of Variation	11.85	15.90	14.27

These reduced data sets will form the basis for the Web proxy workload characterization study undertaken in the next section.

4 Proxy Workload Characterisation

In the following sub-sections, we present a more detailed analysis of Web proxy workload characteristics, including document types and sizes, an analysis of transfer size distributions and their heavy-tails, and an analysis of file referencing behaviour, including document popularity, Zipf-like referencing behaviour, temporal locality, concentration, and "hot set" drift analysis.

4.1 Document Types and Sizes

The next step in our workload characterization study was to classify document requests according to the following generic categories:

- HTML (e.g., .html, .shtml, .htm)
- Image (e.g., .gif, .jpeg, .gif89, .xbm)

- Audio (e.g., .au, .ram, .wav)
- Video (e.g., .mpeg, .avi, .mov)
- Text (e.g., .tex, .readme, .c, .java)
- Compressed (e.g., .zip, .gz)
- Application (e.g., .ps, .pdf, .dvi)
- Dynamic (e.g., .cgi, .perl)

Any document that could not be classified under one of the above categories was placed in the Others category.

The results of this analysis for the USask, CANARIE, and NLANR data sets are summarized in Tables 5, 6, and 7,respectively. These tables show that HTML and Image files account for close to 95% of the total requests. Similar results were reported for client traces by Cunha *et al.* [13] and for server traces by Arlitt and Williamson [3, 4].

The results reported in Tables 5 through 7 are consistent with those reported by Braun and Claffy [8] in their study of the NCSA Web server. That is, most requests are for small(er) files. Arlitt and Williamson [4] also report this as a common Web server workload characteristic.

Unlike Web server workloads, however, Tables 5, 6, and 7 show that Image files are consistently the most requested document type (65-80%), followed by HTML files (17-28%). Similar observations were made by Abdulla *et al.* [1] in their characterization of Web proxy traffic. We also observe that Image files are responsible for the highest percentage of bytes transferred (37-58%), followed by HTML files (16-22%).

A substantial portion of the byte transfer volume is accounted for by Video, Compressed, and Application file types, which despite their relative infrequent access (typically less than 1% of the requests) are large enough to generate 2-10% of the bytes transferred. This observation is substantiated by the large mean and median transfer sizes indicated for these document types, compared to other document types. The COV values within each document type category are much lower than the COV of transfer sizes for the aggregate data set (see Table 4). The large variation in mean transfer sizes across the diverse document types helps to explain the high COV reported in the aggregate data sets.

Table 5: Breakdown of Document Types and Sizes (USask)

Item	HTML	Image	Audio	Video	Text	Compressed	Application	Dynamic	Others
% of Requests	18.97	77.45	0.17	0.03	0.75	0.06	0.59	0.08	1.88
% of Bytes	22.15	57.58	1.48	3.89	1.16	3.12	6.47	0.60	3.53
Mean Transfer Size	9,062	5,770	66,005	875,865	11,957	373,053	84,618	56,613	14,682
Median Transfer Size	4,683	2,263	16,873	275,586	1,526	49,828	5,042	7,321	704
COV of Transfer Size	1.88	6.04	4.05	2.09	9.71	4.60	7.29	6.54	18.60

Table 6: Breakdown of Document Types and Sizes (CANARIE)

Item	HTML	Image	Audio	Video	Text	Compressed	Application	Dynamic	Others
% of Requests	16.83	80.67	0.20	0.05	0.44	0.07	0.45	0.01	1.28
% of Bytes Transferred	16.42	54.35	1.93	5.57	2.73	5.08	7.84	0.01	6.07
Mean Transfer Size	9,276	6,403	89,689	1,165,395	59,027	713,208	163,903	2,089	45,023
Median Transfer Size	4,697	2,573	15,812	49,220	3,215	67,619	5,334	989	1,026
COV of Transfer Size	1.78	2.92	4.80	1.83	6.01	4.38	6.79	2.30	15.42

4.2 One-Time Referencing

A surprising observation made in previous analyses of Web server workloads [3] was that (regardless of the duration of the access log studied) typically 15-30% of the documents accessed in the log were accessed only once in the log. This so-called "one-time" referencing behaviour is of concern for Web caching, since there is clearly no point caching something that will be accessed only once.

The precise cause of this one-time referencing behaviour is not fully understood. Several explanations have been proposed. First, it might indicate the vastness of the World-Wide Web, and the low signal-to-noise ratio for much of its content. Second, it might reflect human nature in browsing habits (e.g., once a site has been visited, there is no need to visit it again). Third, it might reflect the behaviour of content providers, who might use date-based URL names, or who might redesign or modify Web pages on a regular basis to keep them current, while possibly removing or renaming old pages. Fourth, it may reflect the presence of search engines or Web robots that traverse many pages to construct an index. Finally, it may be the consequence of document prefetching (i.e., prediction) algorithms in some proxies and/or client browsers These hypotheses all seem plausible. If any (or all) of them are true, then they indicate a challenging workload environment for

Table 7: Breakdown of Document Types and Sizes (NLANR)

Item	HTML	Image	Audio	Video	Text	Compressed	Application	Dynamic	Others
% of Requests	27.85	67.83	0.24	0.28	0.76	0.32	0.61	0.86	1.26
% of Bytes Transferred	17.05	36.83	0.85	8.63	1.90	8.76	14.73	0.46	10.69
Mean Transfer Size	9,067	8,054	53,004	464,698	36,880	400,739	356,738	7,877	125,678
Median Transfer Size	3,994	2,699	16,697	254,083	2,815	90,730	20,260	2,435	1,068
COV of Transfer Size	2.73	3.51	11.04	2.96	16.38	3.34	5.40	7.03	4.81

Web caching algorithms.

Several other explanations seem less plausible. For example, attributing this one-time referencing to the presence of Dynamic requests (e.g., CGI) is not possible, since Dynamic files typically account for much less than 5% of the workload in Web server and Web proxy access logs. File modification events also tend to occur at low enough rates that they are unlikely to be a factor in most analyses [3]. Attributing one-time referencing to typographical errors made by clients in requested URL names does not make sense, since the access log analyses usually focus on *successful* requests, not errors.

On the positive side, one could argue that one-time referencing occurs because the Web caching hierarchy is working well. That is, repeated requests to the same document are not seen at higher level caches or servers because the document has been pulled into a cache at a lower level (e.g., browser cache, intermediate level proxy cache). While this may well be an indication of (improving) effectiveness of hierarchical Web caching, we remain skeptical, for two reasons. First, the one-time referencing observation, first made in server workloads in 1995, predated much of the large-scale deployment of national-level Web caching hierarchies. Second, the volume of GET If-Modified-Since requests (as indicated by HTTP 304 Not Modified responses) still seems low in many workload studies (e.g., 5-15%), implying either limited effectiveness of caching, or limited cache validation occurring (e.g., lengthy timeouts on cached documents, or browser implementations that don't use or don't support GET If-Modified-Since requests).

In any event, one-time referencing is deemed to be an important workload characteristic of interest. Thus, the next step in our workload characterization study was to assess the one-time referencing behaviour present in Web proxy workloads.

Table 8 summarizes the one-timers with respect to the number of distinct documents and the total requests seen for all the three data sets. We conclude that approximately one-fourth of the total requests (18.1% - 30.3%) are one-timers and approximately 70% of the documents referenced (67.8% - 74.9%) are one-timers. The latter number is significantly higher than the one-timer referencing behaviour seen in Web server workloads [3]. Again, one can only hope that this is an indication that caching hierarchies are actually working as intended.

Table 8: One-Time Referencing Behaviour in Web Proxy Workloads

Item	USask	CANARIE	NLANR
Distinct Documents	5,527,667	1,423,081	7,681,214
One-Timer Documents	???	???	???
One-Timer/Distinct Documents (%)	67.8	74.9	70.9
Total Requests	20,754,720	351,296	20,018,680
One-Timer Requests	???	???	???
One-Timer/Total Requests (%)	18.1	30.4	27.2

Furthermore, the one-time referencing characteristic appears to be independent of (i.e., orthogonal to) document type. Table 9 presents a more detailed analysis of one-timer documents based on document type. This analysis shows that HTML and Image files constitute 95% of the one-timers seen, with each type occuring in the same proportion as they do in the request stream.

Table 9: Breakdown of Document Types for One-Timers

Item	USask	CANARIE	NCSA
HTML	23.0	20.19	22.27
Image	74.38	77.44	74.42
Others	2.62	2.47	3.31
Total	100.0	100.0	100.0

The predominance of one-time referencing for Web documents highlights the need for novel Web caching policies that can effectively discriminate against one-timers. For example, frequency-based algorithms, such as Least

Frequently Used (LFU), tend to perform better than recency-based algorithms, such as Least Recently Used (LRU). Similar observations have been made for Web server caching algorithms [4, 30].

4.3 Transfer Size Distribution

Our next analysis focuses on the transfer size distribution for the documents returned to the requesting clients (either directly by the proxy, or after obtaining the document from a higher-level proxy or the originating server). In particular, we are interested in the shape of this distribution, the presence of heavy-tails in the distribution, and the impact of the heavy-tailed distribution on Web and network performance.

Figure 1 shows the cumulative distribution function of the transfer sizes for each proxy server, using a logarithmic scale on the horizontal axis. Almost all of the transfer sizes are in the range from 100 to 100,000 bytes, with very few small transfers (say, less than 100 bytes) and very few large transfers (say, more than 100,000 bytes). This distribution is similar to the file size distribution reported for Web clients [13] and for Web servers [3, 4, 6, 8, 11].

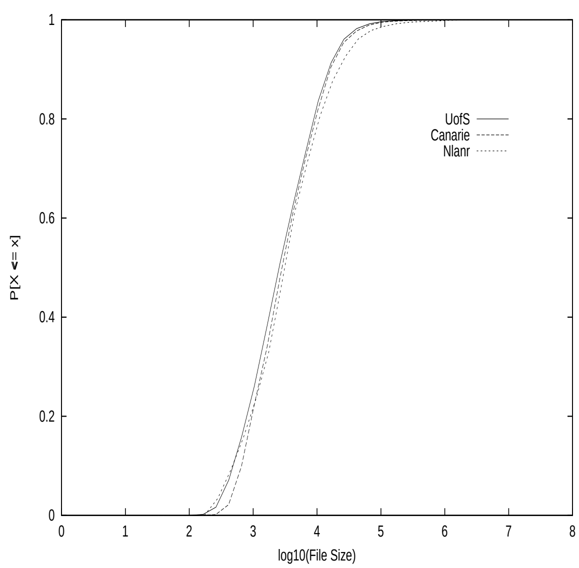


Figure 1: Cumulative Distribution Function for Transfer Sizes, by proxy

The transfer size distribution in Figure 1 is in fact *heavy-tailed.* A distribution is defined to be heavy-tailed if, regardless of the distribution for small values of the random variable, the asymptotic shape of the distribution is hyperbolic [3, 11, 21, 24]. That is, a distribution is heavy-tailed if:

$$P\left[X > x\right] \sim x^{-\alpha}, x \rightarrow \infty, 0 < \alpha < 2$$

The heavy-tailed property of World-Wide-Web workloads is an important characteristic because it has been suggested as one of the causes for the presence of self-similarity (i.e., long-range dependence) in Web traffic [11]. In simple terms, a "heavy tail" to a Web document size distribution means that the very large "elephants" (i.e., outliers) in the tail of the distribution are relatively few in number, but are large enough to contribute significantly to the overall traffic volume observed (e.g., skew the mean transfer size distribution, as observed in Table 4).

The simplest example of a heavy-tailed distribution is the doubly-exponential Pareto distribution; its probability density function is:

$$P\left(x\right) = \alpha k^{\alpha} x^{-\alpha-1}, \alpha, k > 0, x \geq k$$

The cumulative distribution function for the Pareto distribution is:

$$F\left(x\right) = P\left[X \leq x\right] = 1 - \left(k/x\right)^{\alpha}$$

The Pareto distribution has been applied to phenomenon observed in social sciences, such as the distribution of the length of books on a library shelf [22] and distribution of income [18]. This distribution has been used to model FTP data bursts [24, 25] and Web traffic [3, 13, 11].

The α parameter is the known as the tail-index [12], and k defines where the "tail" of the distribution begins (i.e., it represents the smallest possible value of the random variable in the heavy-tailed distribution). As α decreases, the tail of the distribution becomes heavier [11]. In other words, an arbitrarily large portion of the probability mass may be present in the tail of the distribution as α decrease. As k increases, only the tail of the distribution is modelled.

To estimate the tail-index α for our transfer size data sets, we follow the approach outlined in [11] and [5]. First, a log-log complementary distribution (LLCD) is plotted for the transfer sizes in the data sets. A LLCD plot graphs $log\bar{\mathrm{F}}(x) = log(1 - F(x))$ versus $logx$, for large x [5]. Heavy tailed distributions have the following property:

$$\frac{dlog\bar{\mathrm{F}}(x)}{dlogx} = -\alpha, x > k$$

An estimate of the tail-index α is obtained by determining the slope of the LLCD plot for values of x greater than k, using least squares linear regression. Estimates of α can also be obtained using Hill Estimator [32], Maximum Likelihood Estimator [20], Least Square Techniques [24], and Scaling Estimators [12].

Figure 2 shows the LLCD plots for all the data sets, along with the least square regression fits for the heavy-tail ($k = 1000$ bytes). All three data sets show transfer size distributions that are heavy-tailed. Table 10 summarizes the estimated α value for each data set, along with the coefficient of determination (R^2), which assesses the "goodness of fit" for the linear regression. These results show a very strong fit (R^2 is close to 1.0), and α values ranging from 1.1 to 1.3.

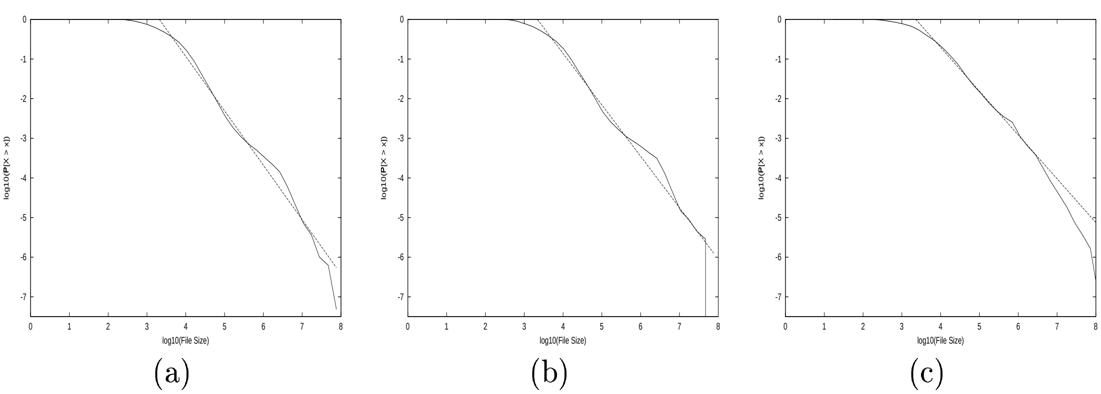
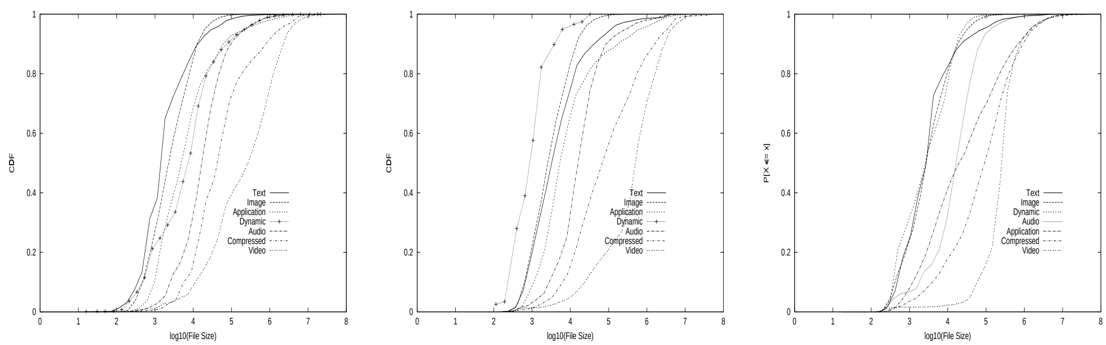


Figure 2: Log-Log Complementary Distribution (LLCD) Plots for Transfer Sizes: (a) USask; (b) CANARIE; (c) NLANR

Among the three data sets considered, the NLANR data set exhibited the most heavy tail ($\alpha = 1.11$), while the USask data set shows the least heavy tail ($\alpha = 1.37$). We conclude that proxy transfer sizes, just like server transfer sizes, are heavy tailed, with α ranging from 1.11 and 1.37.

Table 10: Estimates of α for Heavy-Tailed Transfer Size Distributions

Item	USask	CANARIE	NLANR
α	1.37	1.30	1.11
R^2	0.98	0.98	0.99



(a) (b) (c)

Figure 3: Distribution of File Sizes by Document Type: (a) USask; (b) CANARIE; (c) NLANR

The heavy-tailed nature of the transfer size distributions also applies within each document type category. Figure 3 shows the cumulative distribution function for transfer sizes by document type, for the three data sets. These distributions all show heavy-tail behaviour consistent with the foregoing analyses.

4.4 Document Popularity versus Document Size

It was observed earlier that most Web transfers involve small documents, as reflected by a median value lower than the mean value of the transfer size. To determine whether there is any relationship between the frequency of reference to a document and the document size, a simple visual test was performed.

Figure 4 shows the frequency of access versus transfer size on a log-log plot. Visual inspection of the plots suggests that small documents are transferred more often than large documents. To ascertain this claim, the correlation coefficient is calculated for each of the data sets. The correlation coefficient always lies between -1 and +1: a correlation coefficient close to +1 would indicate that large transfers occur more often, whereas a correlation coefficient close to -1 indicates that large transfers occur less often. A correlation coefficient close to zero indicates no linear relationship between number of requests and document size.

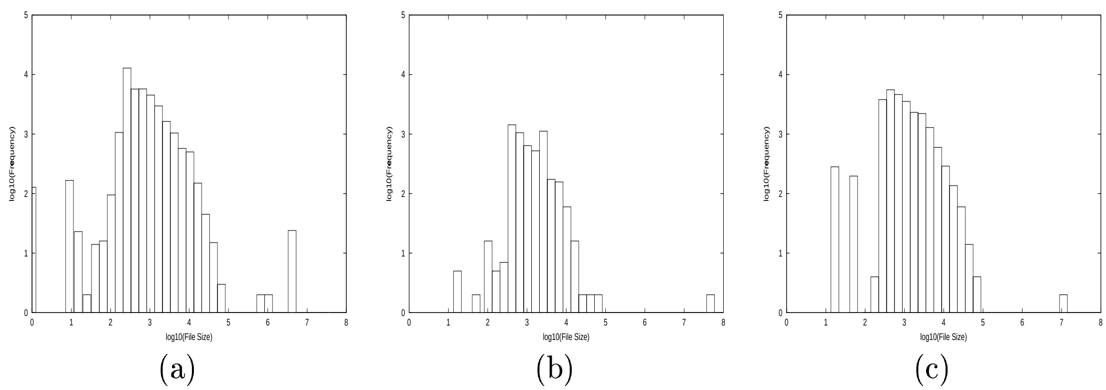


Figure 4: Frequency of Access versus Transfer Size: (a) USask; (b) CANARIE; (c) NLANR

The correlation coefficients for each of the data sets are shown in Table 11. The results indicate a strong negative correlation between the frequency of access to documents and the transfer sizes. That is, transfers of large documents occur less often. The correlation coefficients for individual file types for each data set was also calculated (results not given here). It was observed that the Image files and the HTML files have a very high negative correlation coefficient, while the Video and Compressed files had little correlation between transfer size and frequency of access. This might be because of browsing habits of users. Users like to download web documents on the click of the mouse button, which is possible for small documents. Therefore, there are more accesses to smaller sized HTML and Image files. While downloading Video or Compressed files, users have less choice because most of these documents are large; hence there seems to be no correlation between the transfer sizes and the frequency of reference.

Table 11: Correlation Analysis: Frequency of Reference versus Transfer Size

Item	USask	CANARIE	NLANR
Correlation Coefficient	-0.84	-0.84	-0.83

4.5 File Referencing Behaviour

This section studies various characteristics of the file referencing behaviour in Web proxy workloads.

4.5.1 Document Popularity

The highly uneven popularity of Web documents has been noticed by many researchers [2, 4, 6, 5, 8, 9, 13], and Zipf's Law [33] has been applied to model this behaviour. Zipf's Law states that if items are ranked (r) according to their popularity (P), then the popularity of an item is inversely proportional to its rank.

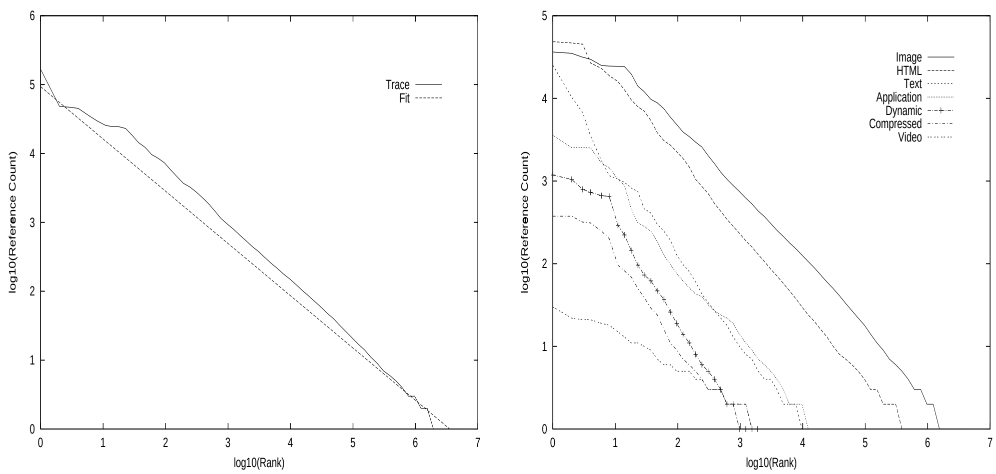
The Zipf distribution is a parameter-less hyperbolic distribution of the form:

$$P \sim 1/r$$

Note that in this distribution, r is raised (exactly) to the power -1. In other words, the N^{th} most most popular item is exactly twice as popular as the $2N^{th}$ most popular item, and so on [2]. This type of referencing behaviour is prevalent in many information systems (e.g., memory referencing behaviour of computer programs[10, 26], popularity of words used in the English language, popularity of books borrowed from a public libary [22, 33], movies rented from a video store).

In the present context, the Zipf distribution implies that a few documents are very highly referenced, a moderate number of documents are moderately referenced, while a large number of documents are referenced only a few times. To see whether or not documents follow the Zipf distribution, the documents are first sorted in descending order according to their frequency of reference. The documents are then ranked, with the most referenced document being assigned a rank of one, followed by the next most referenced document with a rank of two, and so on.

The frequency versus rank plots for the three data sets appear in Figures 5, 6, and 7 for the USask, CANARIE, and NLANR data sets, respectively. In each of these figures, the graph on the left shows the referencing behaviour for *all* documents in the data set, while the graph on the right refines the analysis to each category of document type.



(a) (b)

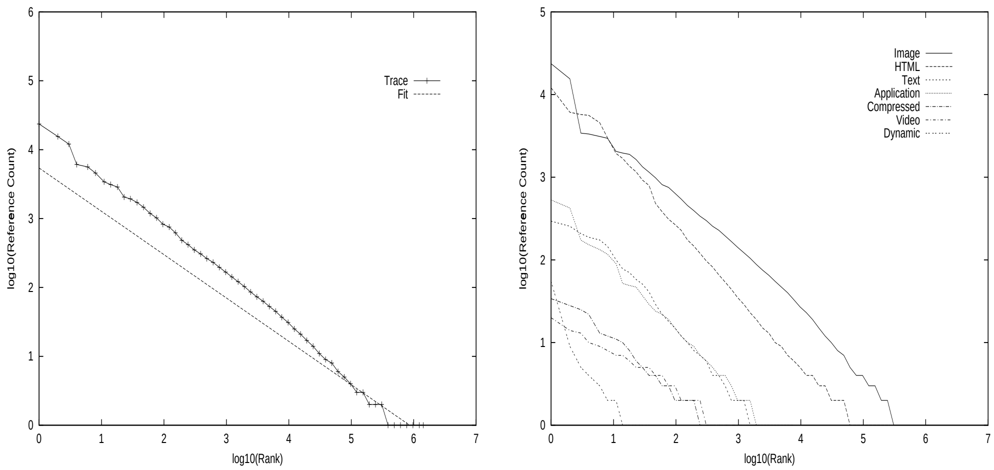
Figure 5: Reference Count versus Rank (USask): (a) All Files (b) By File Type

Visual inspection suggests that the referencing behaviour in all cases follows a Zipf-like distribution, where:

$$P = k\left(\frac{1}{r}\right)^{-\beta}$$

To determine the slope (β) of the aggregate data set (all files), a least squares fit over the (log-transformed) data set is performed.

The calculated values for β and the goodness of fit (R^2) values are shown in Table 12. It is evident that the referencing behaviour is *not* Zipfian in nature, as shown by the lower β values. It can also be observed that the Least-Squares fit and the empirical distribution do not agree very strongly; this is primarily due to the presence of many documents of low popularity (i.e., more than predicted by Zipf's Law) and many one-timers. We thus conclude that document popularity follows a Zipf-like referencing behaviour, but does not precisely match a Zipf distribution.



(a) (b)

Figure 6: Reference Count versus Rank (CANARIE): (a) All Files (b) By File Type

Table 12: Estimated Slopes for Zipf-Like Referencing Distribution

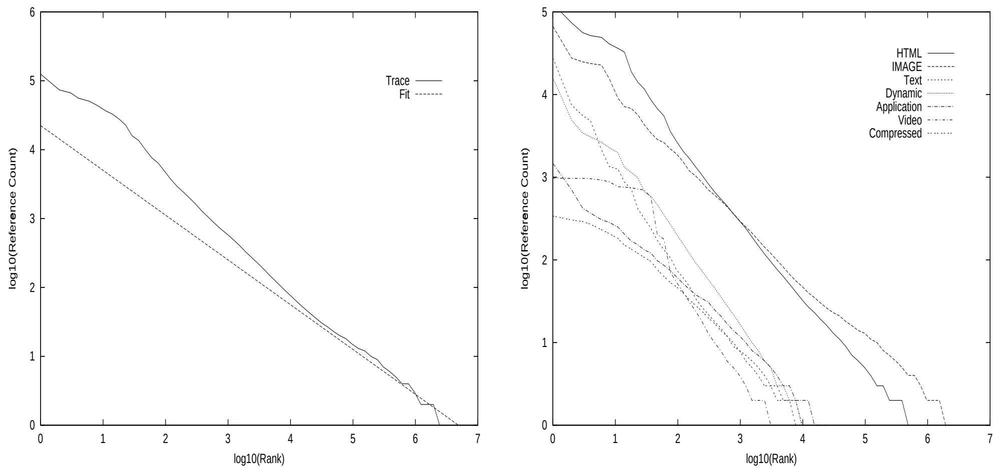
Item	USask	CANARIE	NLANR
β	0.76	0.63	0.65
R^2	0.93	0.88	0.89

4.5.2 Concentration of References

Another way of characterising the uneven document access patterns is to determine the extent to which references are skewed towards certain documents. This measure of skewness, referred to as *concentration*, was originally applied to file referencing behaviour in an Unix environment [10, 31], and later to Web server document accesses [4, 8, 19].

The concentration phenomenon is illustrated in Figures 8, 9, and 10 for the USask, CANARIE, and NLANR data sets, respectively. Again, two graphs are used in each figure: the graph on the left shows the overall document referencing concentration, and the graph on the right refines the analysis based on document types.

Non-uniform referencing of Web documents is clearly reflected in these

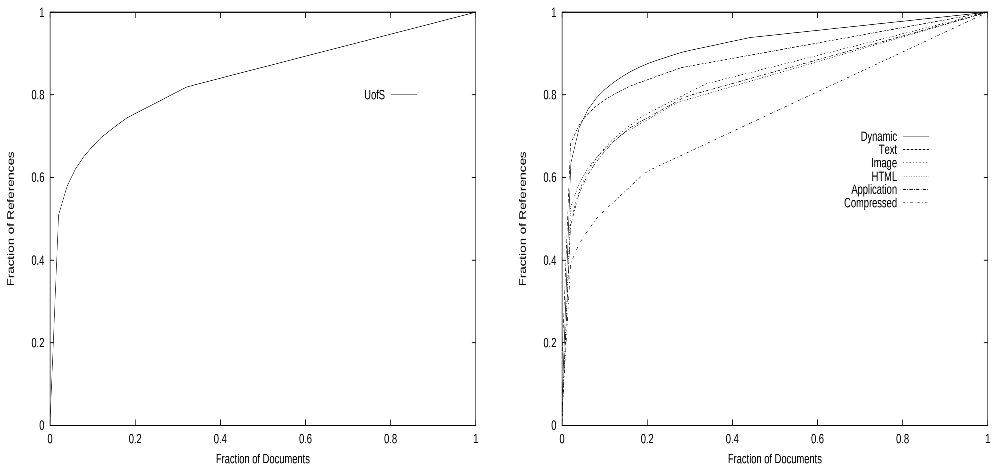


(a) (b)

Figure 7: Reference Count versus Rank (NLANR): (a) All Files (b) By File Type

Figures. The USask data set shows the most concentration of references among the three data sets, with approximately 30% of the documents accounting for about 80% of the references. The remaining 70% of the documents (primarily the one-timers) account for the remaining 20% of the requests.

These results suggest that concentration of references is lower at Web proxy servers than at Web servers [3, 4]. This observation makes sense intuitively, since clients at a Web proxy can effectively access any available document in the web (i.e. the document set is very large). For Web servers, the requests are restricted to a limited set of documents (i.e. the documents present in the Web server). Therefore, it is almost natural to observe more concentration of references at the servers compared to the proxies. We can also hypothize that a proxy server with homogeneous clients (institutional level proxy caches like the USask cache) can expect to observe more concentration of references compared to that at higher level caches (like the CANARIE and NLANR caches).



(a) (b)

Figure 8: Concentration of References (USask): (a) All Files (b) By File Type

4.5.3 Temporal Locality

Temporal locality refers to the property that an object referenced in the recent past will likely be referenced again in the near future. To measure temporal locality, the Least Recently Used Stack Model (LRUSM) is used [27]. The LRUSM is a stack-based ordering of referenced objects, according to their recency of reference (i.e., most recently referenced item on top (position 1), and the least recently referenced item on the bottom). For each reference in the request stream, the stack is searched until the requested object is found, or the bottom of the stack is reached. If found (i.e., a hit), the object is removed from its present position in the stack (say, d) and moved to the top of the stack, pushing the other $d-1$ items that used to be above it down one position, as needed. For an item that is not found in the stack (i.e., a miss), it is simply added to the top of the stack, pushing all other stack items down one level. The most important part of the LRUSM is keeping track of the stack depth d at which each hit occurs. The presence of temporal locality manifests itself in a large number of hits at or near the top of the stack.

Figure 11 shows the stack depth referencing frequency for all the data sets

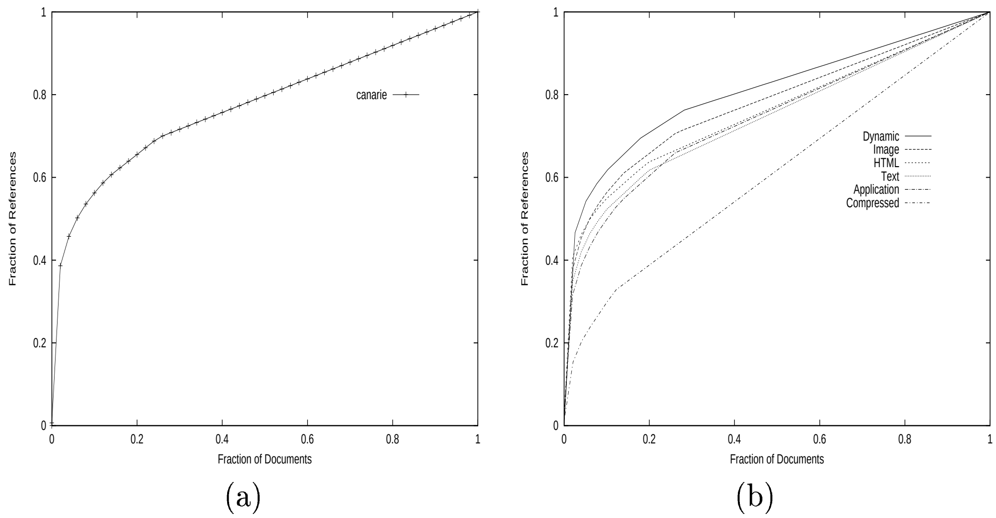


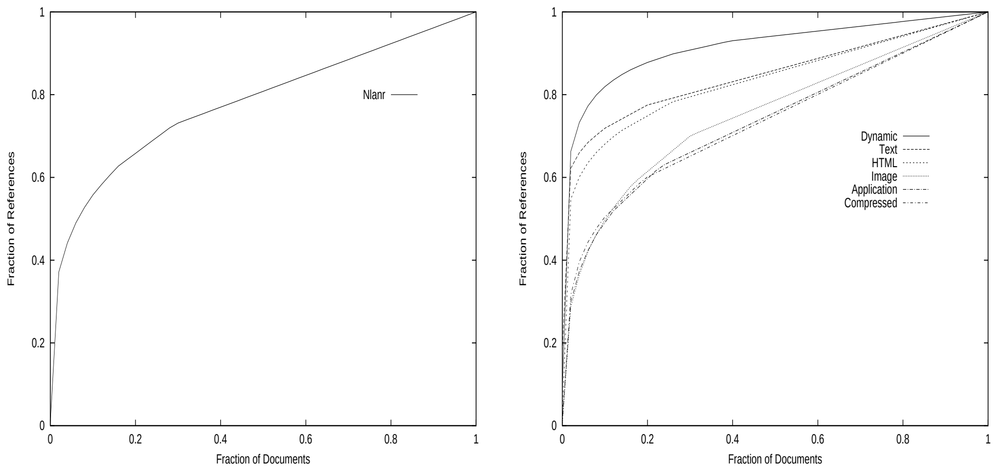
Figure 9: Concentration of References (CANARIE): (a) All Files (b) By File Type

up to a stack depth of 100. It is immediately obvious that the USask data set exhibits more temporal locality than the CANARIE and NLANR data sets. This is due to the fact that both CANARIE and NLANR proxies are higher-level caches which receive filtered requests (i.e., requests that lower level caches could not service), from their clients, which are predominantly lower level proxies.

We also note that temporal locality at the proxy servers, in general, is not very high (compared to temporal locality results for Web servers [3]). This may be due to caching effects elsewhere in the hierarchy, or due to the document referencing behaviour generated by clients.

4.5.4 "Hot Set" Drift Analysis

A simple analysis was performed to understand how the "hot set" of documents changes with time for each Web proxy. For each of the data sets, the most popular 500 documents were found on the inital day of the traces. Then the most popular 500 documents were found for each of the following days in the access log. The overlap of the documents in these hot sets for each of the following days in the trace with respect to the starting day provides a



(a) (b)

Figure 10: Concentration of References (NLANR): (a) All Files (b) By File Type

measure of the "drift" in the hot set.

Figure 12 shows the results of the hot set analysis. The USask data set shows the least drift in the hot set, while the CANARIE data set shows the most drift in the hot set. The slow decay of the USask hot set shows that some documents have long term popularity. The fast decay of the CANARIE hot set suggests that the lower level caches tend to cache the popular documents effectively. The intermediate behaviour of the NLANR data set implies that there are some documents whose popularity at the proxy cache is due to the multiplexed references from various other lower level caches. In other words, certain documents are popular at many lower level cahes, but not popular enough to make their way into the lower level proxy cache.

5 Conclusions

This document has reported on workload analysis of Web proxy access logs from three different sites in a large-scale Web caching hierarchy.

The main observations from our study are:

- HTML and Image files account for 95% of the total requests.

- Most Web document transfers are small. The mean document transfer size is 7-15 KB, while the median transfer size is 2-3 KB.
- Transfer size distributions are heavy-tailed.
- Approximately one-fourth of the total requests are one-timers and approximately 70% of the documents referenced are one-timers.
- The popularity of Web documents does *not* strictly follow Zipf's law, but it does follow a Zipf-like referencing distribution.
- The concentration of requests is higher at lower-level Web proxies than at higher-level proxies.
- Temporal locality in the document request stream at Web proxy servers is generally low.

Work is on progress to characterise inter-arrival of document requests, quantifying temporal locality, and ascertaining the applicability of the Independence Reference Model (IRM) [14, 15] to model document references for Web proxy servers.

References

[1] Ghaleb Abdulla, Edward A. Fox, Marc Abrams, and Stephen Williams. Www proxy traffic characterisation with application to caching. Technical Report TR-97-03, Virginia Tech. department of Computer Science, March 1997.

[2] V. Almeida, Azer Bestovros, Mark Crovella, and Adriana de Oliveira. Characterizing reference locality in the www. In *Proceedings of 1996 International Conference on Parallel and Distributed Information Systems (PDIS '96)*, pages 92–103, December 1996.

[3] Martin F. Arlitt. A performance study of web servers. Master's thesis, University of Saskatchewan, 1996.

[4] Martin F. Arlitt and Carey L. Williamson. Internet web servers: Workload characterisation and performance implications. *IEEE/ACM Transactions On Networking*, 5(5):631–645, October 1997.

[5] Paul Barford, Azer Bestavros, Adam Bradley, and mark Crovella. Changes in web client access patterns: Characteristics and caching implications. Technical Report BUCS-TR-1998-023, Boston University Department of Computer Science, November 1998.

[6] Paul Barford and Mark Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of Performance '98/SIGMETRICS '98*, pages 151–160, July 1998.

[7] Dave Bickle. Personal communication (e-mail).

[8] Hans-Werner Braun and Kimberly C. Claffy. Web traffic characterisation: an assessment of the impact of caching documents from ncsa's web server. *Computer Networks and ISDN Systems*, 28:37–51, 1995.

[9] L. Breslau, P. Cao, L. Fan, g. phillips, and S. shenkar. On the implications of zipf's law for web caching. In *Proceedings of Infocom'99*, 1999.

[10] R. B. Bunt and J. M. Murphy. The measurement of locality and the behaviour of programs. *The Computer Journal*, 27(3), 1984.

[11] Mark Crovella and Azer Bestavros. Self-similarity in world wide web traffic: Evidence and possible cause. *IEEE/ACM Transcations on Networking*, 5(6):835–846, December 1997.

[12] Mark E. Crovella and Murad S. Taqqu. Estimating the heavy tail index from scaling properties. *Methodology and Computing in Applied Probability*, 1(1), 1999.

[13] Carlos A. Cunha, Azer Bestavros, and Mark Crovella. Characteristics of www client-based traces. Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.

[14] P. Denning and S. Schwartz. Properties of the working set model. *Communications of the ACM*, 15(3):191–198, 1972.

[15] Peter J. Denning. Working sets past and present. *IEEE Transcations on Software Engineering*, 6(1):64–84, January 1980.

[16] Eric Fogel. Personal communication (e-mail).

[17] Hardware for the caches. http://www.ircache.net/Cache/hardware.html.

[18] N. Jhonson and S. Kotz, editors. *Encyclopedia of Statistical Sciences*, volume 6 and 9. John Wiley & Sons, Inc., New York, NY, 1988.

[19] T. Kwan, R. McGrath, and D. Reed. Ncsa's world wide web server: Design and performance. *IEEE Computer*, 28(11):68–74, November 1995.

[20] Averill M. Law and W. David Kelton. *Simulation Modelling and Analysis*. McGraw Hill Book Co., 1982.

[21] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–14, February 1994.

[22] Benoit B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freedman and Co., New York, 1983.

[23] Nlanr cache status. http://www.ircache.net/Cache/status.html. Squid version.

[24] Vern Paxson. Empirically derived analytic models of wide-area tcp connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.

[25] Vern Paxson and Sally Floyd. Wide-are traffic: The failure of poisson modelling. In *Proceedings of ACM SIGCOMM '94*, pages 257–268, London, England, August 1994.

[26] J. Peachey. The bradford-zipf distribution and program behaviour. Master's thesis, University of Saskatchewan, Department of Computational Science, 1982.

[27] Jeffery Spirn. Distance string models for program behaviour. *IEEE Computer*, 13(11), November 1976.

[28] J. Vass, J. Harwell, H. Bharadvaj, and A. Joshi. The world wide web:everything you (n)ever wanted to know about its servers. *IEEE Potentials*, pages 33–34, October/November 1998.

[29] Duane Wessels. Squid frequently asked questions. http://squid.nlanr.net/Squid/FAQ/FAQ.html.

[30] Stephen Williams, Marc Abrams, Charles R. Standridge, Ghaleb Abdulla, and Edward A. Fox. Removal policies in network caches for world-wide web documents. In *Proceedings of ACM SIGCOMM '96*, 1996.

[31] Carey L. Williamson and R. B. Bunt. Characterising short term file referencing behaviour. Technical report, Department of Computational Science University of Saskatchewan, Saskatoon, Canada, December 1985.

[32] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. In *Proceedings of SIGCOMM '95*, pages 100–113, Boston, MA, 1995.

[33] G. K. Zipf. *Human Behavoiur and the Principle of Least-Effort.* Addison-Wesley, Cambridge, MA, 1949.

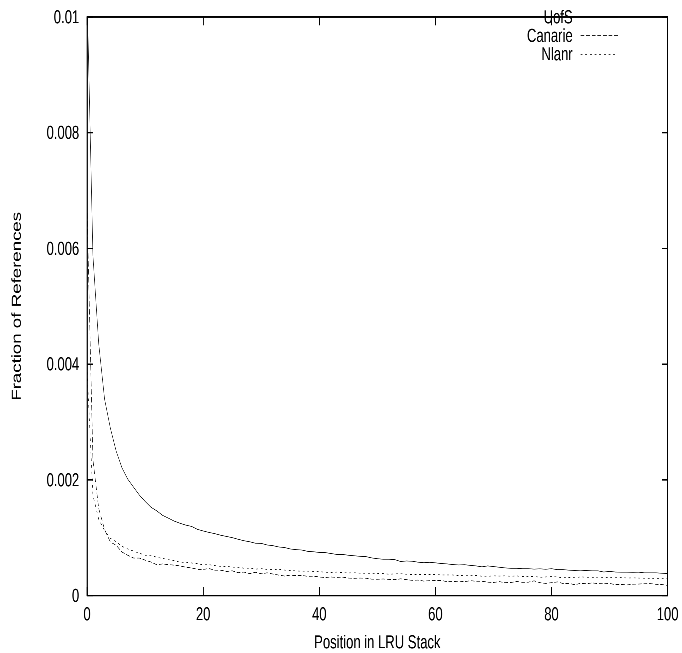


Figure 11: Temporal locality characteristics for three data sets

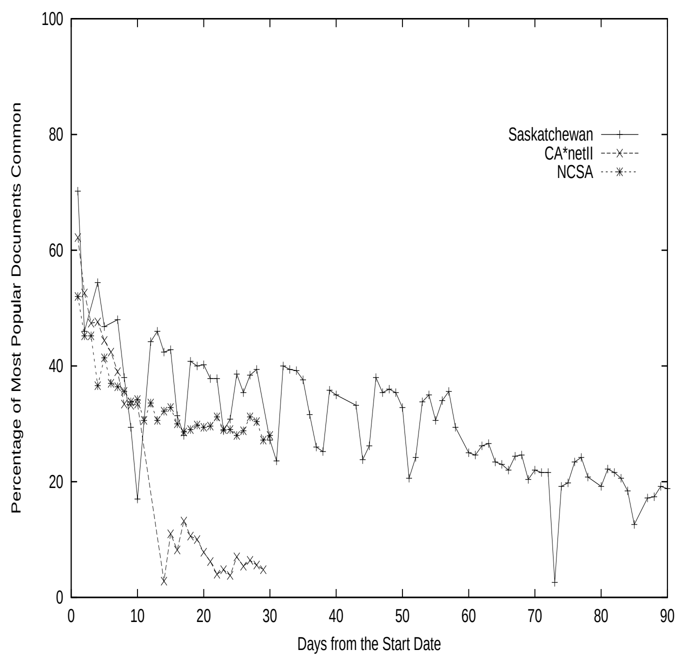


Figure 12: Hot Set Drift