Internet Web Servers: Workload Characterization and Performance Implications

Martin F. Arlitt Carey L. Williamson

Abstract—

This paper presents a workload characterization study for Internet Web servers. Six different data sets are used in the study: three from academic environments, two from scientific research organizations, and one from a commercial Internet provider. These data sets represent three different orders of magnitude in server activity, and two different orders of magnitude in time duration, ranging from one week of activity to one year of activity.

The workload characterization focuses primarily on the document type distribution, the document size distribution, the document referencing behaviour, and the geographic distribution of server requests. Throughout the study, emphasis is placed on finding workload characteristics that are common across all the data sets studied. Ten such characteristics are identified. The paper concludes with a discussion of caching and performance issues, using the observed workload characteristics to suggest performance enhancements that seem promising for Internet Web servers.

1 Introduction

The popularity of the World-Wide Web [7, 37] (also called WWW, or the Web) has increased dramatically in the past few years. In December 1992, WWW traffic was almost non-existent (only 74 MB per month on the NSFNET network backbone [26]). Today, WWW traffic is one of the dominating components of Internet traffic.

There are many reasons behind this explosive growth in Web traffic. These reasons include: the ease of use of the Web; the availability of graphical user interfaces for navigating the Web; the availability of editors and support tools for creating and "publishing" Web documents; an emerging trend among researchers, educational institutions, and commercial organizations to make the Web the standard mechanism for disseminating information in a timely fashion; the machine-independent nature of the languages and protocols used for constructing and exchanging Web documents; and a continuing exponential increase in the number of Internet hosts and users [25, 30].

The phenomenal and alarming growth in Web traffic has sparked much research activity on "improving" the World-Wide Web (see Section 2.3). Much of this recent research activity has been aimed at improving Web performance and scalability. The key performance factors to consider are how to reduce the volume of network traffic produced by Web clients and servers, and how to improve the response time for WWW users.

Martin Arlitt and Carey Williamson are with the Department of Computer Science, University of Saskatchewan. A preliminary version of this paper appeared in [5].

Fundamental to the goal of improving Web performance is a solid understanding of WWW workloads. While there are several studies reported in the literature [8, 10, 11, 12, 20], most studies present data from only one measurement site, making it difficult to generalize results to other sites. Furthermore, some studies focus on characterizing Web clients and Web proxies, rather than Web servers.

The purpose of this paper is to present a detailed workload characterization study of Internet Web servers, similar to earlier studies of wide-area network TCP/IP traffic [14]. Six different Web server access logs are used in this study: three from academic environments, two from scientific research institutions, and one from a commercial Internet provider. The data sets represent three different orders of magnitude in server activity, ranging from 653 requests per day to 355,787 requests per day, and time durations ranging from one week of activity to one year of activity.

Throughout the study, emphasis is placed on finding workload characteristics that are common across all the data sets studied. These characteristics are deemed important since they potentially represent universal truths for all Internet Web servers. Our research has identified ten such characteristics for Web server workloads. These characteristics are summarized in Table 1, for easy reference, and are described in more detail within the paper itself.

The remainder of this paper is organized as follows. Section 2 provides background material on the World Wide Web, and a discussion of related work. Section 3 describes the Web server logs used in this study, and presents summary statistics for the six data sets. Section 4 presents the detailed results of our workload characterization study. The paper concludes, in Section 5, with a discussion of caching and performance issues for Internet Web servers, drawing upon the observed workload characteristics to identify performance enhancements that seem promising for Internet Web servers.

2 The World Wide Web

2.1 Web Overview

The Web is based on the client-server model [33, 38]. Communication is always in the form of request-response pairs, and is always initiated by the client.

A client accesses documents on the Web using a Web browser [37]. When the user selects a document to retrieve, the browser creates a request to be sent to the correspond-

Table 1: Summary of Workload Characteristics Common to Internet Web Servers

Characteristic	Name	Description
1 (Section 3.3)	Successful Requests	Approximately 80-90% of the requests to a Web server result in the successful return of a document (file)
2 (Section 3.5)	Document Types	HTML and image documents together account for over 90% of the documents transferred by Web servers
3 (Section 3.5)	Median Transfer Size	The median transfer size for Web server documents is small (e.g., less than 5 kilobytes)
4 (Section 3.5)	Distinct Requests	Among all server requests, very few (e.g., 1-3%) of the requests are for separate (distinct) documents
5 (Section 3.5)	One Time Referencing	A significant percentage (e.g., 15-40%) of the files and bytes accessed in the log are accessed only once in the log
6 (Section 4.1)	File Size Distribution	The file size distribution and the transfer size distribution are *heavy-tailed* (e.g., Pareto with $\alpha \approx 1$)
7 (Section 4.2.1)	Concentration of References	10% of the files accessed on the server typically account for 90% of the server requests and 90% of the bytes transferred
8 (Section 4.2.3)	Inter-Reference Times	Successive references to the same file are exponentially distributed and independent
9 (Section 4.2.4)	Remote Requests	Remote sites account for most (e.g., $\geq 70\%$) of the accesses to the server, and most (e.g., $\geq 60\%$) of the bytes transferred
10 (Section 4.2.4)	Wide Area Usage	Web servers are accessed by hosts on many networks, with 10% of the networks accounting for most (e.g., $\geq 75\%$) of the usage

ing Web server. Each Web *page* may consist of multiple documents (files). Currently, each file is requested separately from the server.

A Web server responds to each request it receives from Web clients. The response from the server includes a status code to inform the client if the request succeeded. If the request was successful, then the response includes the requested document. If the request was unsuccessful, a reason for the failure is returned to the client [27, 38].

2.2 Recording Web Server Workloads

Web servers can be configured to record (in an *access* log) information about all of the requests and responses processed by the server [24]. Each line from the access log contains information on a single request for a document. The log entry for a normal request is of the form:

```
hostname - - [dd/mmm/yyyy:hh:mm:ss tz] request status bytes
```

From each log entry, it is possible to determine the name of the host machine making the request, the time that the request was made, and the name of the requested document. The entry also provides information about the server's response to this request, such as if the server was able to satisfy the request (if not, a reason why the response was unsuccessful is given) and the number of bytes transmitted by the server, if any.

The access logs provide most of the data needed for workload characterization studies of Web servers. However, they do not provide *all* of the information that is of interest. For example, the log entries tell only the number of bytes transferred for a document, not its *actual size*[1]; there is no record of the *elapsed time* required for a document transfer; and there is no information on the *complete set of files* available on the server, other than those documents that are accessed in the logs. Furthermore, there is no record of whether a file access was human-initiated or software-initiated (e.g., by a *Web crawler* [2]), or what caching mechanisms, if any, are in place at the client and/or the server. These issues are outside the control of our study: our focus is solely on characterizing the workload seen by a typical Internet Web server in its de facto configuration.

2.3 Performance Issues and Related Work

The phenomenal growth in Web traffic has led to many performance problems, which in turn has resulted in much research activity on "improving" the World-Wide Web. The overall performance of the Web is determined by the performance of the components which make up the Web: the clients, the servers, the proxies, the networks, and the protocols used for communication.

Efficient Web browsers (clients) can use file caching to reduce the loads that they put on Web servers and network links. A recent study at Boston University [8] evaluated the effects of client-level caching on Web performance.

Improving the performance of Web servers is vital to the goal of reducing response times. Researchers at Boston University are developing a tool that takes detailed measurements of Web server activity, to help identify perfor-

[1] These two values can differ. For example, in most Web browsers, users can abort a document transfer at any time, making the transfer size reported in the log smaller than the actual document size.

[2] A Web crawler tends to visit a large number of documents in a very short period of time, while a human user typically visits a smaller set of documents, usually with longer inter-reference times (corresponding to user "think times").

mance bottlenecks [2]. Yeager and McGrath [38] evaluate the performance impact of different Web server designs. Other researchers have studied the use of file caching in reducing Web server loads [21].

Web proxies are useful for reducing response times and network traffic [16]. Researchers at several institutions are studying various cache replacement policies for Web proxies [1, 9, 36].

Several studies have suggested the use of network file caches to reduce the volume of traffic on the Internet [10, 13]. Researchers at NLANR have implemented a prototype hierarchy of caches, and are currently focusing on configuring and tuning caches within the global hierarchy [34].

The current protocol used for client-server interaction within the World-Wide Web (i.e., HTTP) is quite inefficient. A more efficient approach would allow for multiple client requests to be sent over a single TCP connection [23, 27].

Spasojevic *et al.* [32] suggest using a wide-area file system within the World-Wide Web. Current filesystems, such as AFS [17], have mechanisms to address performance, reliability and security, problems with which the World-Wide Web is currently struggling.

Although the primary focus of this paper is workload characterization for Web servers, several relevant issues affecting server caching and performance are discussed in Section 5. Client, proxy, network and protocol performance issues are outside the scope of this paper.

3 Data Collection and Analysis

This section presents an overview of the six separate data sets used in our workload characterization study. Section 3.1 describes the data collection sites, Sections 3.2 and 3.3 present the "raw" log contents, Section 3.4 discusses the reduction of the raw data from the access logs into more manageable form, Section 3.5 analyzes document types and sizes, and Section 3.6 summarizes the statistical characteristics of the six data sets.

3.1 Data Collection Sites

The access logs used in this research were obtained from six World Wide Web servers: a lab-level Web server at the University of Waterloo (Shoshin Research Lab, Department of Computer Science); a department-level Web server at the University of Calgary (Department of Computer Science); a campus-wide Web server at the University of Saskatchewan; the Web server at NASA's Kennedy Space Center; the Web server from ClarkNet, a commercial Internet provider in the Baltimore - Washington D.C. region; and the Web server at the National Center for Supercomputing Applications (NCSA) in Urbana-Champaign, Illinois.

3.2 Raw Data

Table 2 summarizes the raw data from the six access logs. For ease of reference, the sites are presented in increasing order of server activity, based on the number of requests per day. The same ordering is maintained in all tables throughout the paper.

The six access logs provide information on servers with very different workloads. Table 2 shows that the Waterloo server had a very light workload, while the Saskatchewan server had an order of magnitude more requests to handle. The ClarkNet and NCSA servers had much heavier workloads, more than an order of magnitude greater than the Saskatchewan server. The level of server activity represented in the six logs varies by almost three orders of magnitude, so that our workload characterization study covers light, medium, and heavy workloads. The logs also span different time durations, so that we can study short term, medium term, and long term aspects of Web server file referencing activity.

3.3 Access Log Analysis

The first step in our data analysis was to study the response codes in the Web server access logs. There are many possible responses to client requests. These include: (1) *Successful*: a valid document, which the client has permission to access, was found on the server and returned to the client (or partially returned, if the client aborted); (2) *Not Modified*: the client, which already has a copy of the document in its cache but wishes to verify that the document is up-to-date, is told that the document has not been modified at the server (thus no data bytes need to be transferred); (3) *Found*: the requested document is known to reside in a different location than was specified by the URL provided by the client, so the server responds with the new URL (but not the document); and (4) *Unsuccessful*: either no such document exists, the client did not have permission to access this document, or an error occurred (at the server or during network communication).

Table 3 provides an overall view of the response code frequencies observed in the access logs. From Table 3, we can identify the first common characteristic in Web server workloads. *Successful* responses made up 78-92% of all responses in the logs. Cache related queries that result in *Not Modified* account for 4-14%. The latter observation suggests either the limited use or the limited effectiveness of client-side or network-level caching in the World-Wide Web. The *Successful* and *Not Modified* requests account for 92-97% of all requests across the six data sets.

3.4 Data Reduction

Since the *Successful* responses account for all of the documents transferred by the server, only these responses are used for the remaining analyses in this paper. This simplification focuses the workload characterization on the most common events.

Table 4 provides a statistical summary of the reduced

Table 2: Summary of Access Log Characteristics (Raw Data)

Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Access Log Duration	8 months	1 year	7 months	2 months	2 weeks	1 week
Access Log Start Date	Jan 1/95	Oct 24/94	Jun 1/95	Jul 1/95	Aug 28/95	Aug 28/95
Total Requests	158,601	726,739	2,408,625	3,461,612	3,328,632	2,490,512
Avg Requests/Day	653	2,059	11,255	56,748	237,759	355,787
Total Bytes Transferred (MB)	1,701	7,577	12,330	62,483	27,592	28,268
Avg Bytes/Day (MB)	7.0	21.5	57.6	1,024.3	1,970.9	4,038.3

Table 3: Breakdown of Server Responses for All Data Sets

Response Code	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Successful	86.6%	78.1%	90.7%	89.3%	88.4%	92.0%
Not Modified	7.8%	13.5%	6.3%	7.7%	8.1%	4.1%
Found	1.7%	4.2%	1.7%	2.1%	0.9%	0.3%
Unsuccessful	3.9%	4.2%	1.3%	0.9%	2.6%	3.6%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

data sets. This table shows that the number of distinct documents requested from the server is significantly lower than the total number of documents requested, implying that some documents are requested many, many times. The mean size of the documents transferred is quite small ($\approx$ 5-21 Kbytes), as might be expected, and the median is even smaller ($\approx$ 2-4 Kbytes). These relatively small sizes suggest that Web users (and Web page designers) are very conscious of the impact of document size on download time, particularly over low speed Internet links.

3.5 Document Types and Sizes

The next step in our analysis was to classify documents by type, using the generic categories HTML, Images, Sound, Video, Formatted, and Dynamic files. Classification was based on the suffix used in file names (e.g., `.html`, `.gif`, `.au`, `.mpeg`, `.ps`, `.cgi`, and many more). Unrecognized document types are classified as Other.

For each of the data sets in Table 4, statistics on the type of document requested were calculated. The results from this analysis for two of the six data sets (NCSA and NASA) are shown in Figure 1. The graphs show the percentage of each document type seen based on requests, bytes transferred, distinct files accessed on the server, and distinct bytes accessed on the server.

The document type analysis identified a second common characteristic in Web server workloads. Across the six data sets, HTML and Image documents accounted for 90-100% of the total requests to the server.[3] This observation is consistent with results reported by Sedayao [31] and by Cunha, Bestavros and Crovella [12]. Both of these papers reported that over 90% of client requests were for either HTML or image documents.

Table 4 also indicates that most transferred documents are quite small, which is a third common characteristic. This phenomenon was also seen by Braun and Claffy [10] for requests to the NCSA's Web server. Despite the fact that Web browsers provide support for the use of multimedia objects like sound and video, documents of these types accounted for only 0.01-1.2% of the requests in the six data sets. However, these types of files account for 0.2-30.8% of the bytes transferred, since these files tend to be much larger than other file types. Future growth in the use of video and audio files, made even easier with tools like CGI scripts [38] and Java [18], may have a dramatic impact on Web server workloads.

Finally, Table 5 presents a breakdown of the distinct documents requested from each server. Distinct documents are determined based on the URLs in the access log.

Table 5 illustrates two additional workload characteristics. First, only 0.3-2.5% of the requests and 0.3-6.0% of the bytes transferred are for distinct documents. This observation implies that caching documents (at the server, at the client, or within the network) could greatly improve the performance of the server, as has been pointed out by Claffy and Braun [10]. Second, in all six data sets, approximately one-third (e.g., 22.6-42.1%) of all the distinct documents are requested only once, and one-third (e.g., 15.3-39.5%) of the distinct bytes are transferred only once. This observation is somewhat surprising given that the six data sets represent time durations ranging from one week to one year. This "one time" referencing behaviour has obvious implications on the maximum possible effectiveness of document caching policies. Further discussion of these implications is deferred until Section 5.

3.6 Summary

This section has summarized the statistical characteristics of the six data sets used for our workload characterization study. While the six access logs differ greatly in duration and server activity, five common workload characteristics have been identified. These are summarized in the first five rows of Table 1. The next section examines file referencing patterns and file size distributions for Internet Web servers, looking for further workload characteristics.

[3] In our data sets, there is no common characteristic for HTML documents alone, or for Image documents alone. In fact, the usage of HTML and Image document types differs quite significantly for the NCSA and NASA data sets illustrated in Figure 1.

Table 4: Summary of Access Log Characteristics (Reduced Data, Successful Requests Only)

Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Total Requests	137,277	567,794	2,184,535	3,092,291	2,940,873	2,290,299
Avg Requests/Day	565	1,608	10,208	50,693	210,062	327,186
Distinct Requests	3,413	8,369	18,871	9,355	32,240	23,864
Distinct Requests/Day	14	24	88	153	2,303	3,409
Total Bytes Transferred (MB)	1,701	7,577	12,330	62,483	27,592	28,268
Avg Bytes/Day (MB)	7.0	21.5	57.6	1,024.3	1,970.9	4,038.3
Total Distinct Bytes (MB)	103.0	264.3	249.2	204.7	414.9	666.6
Distinct Bytes/Day (MB)	0.42	0.75	1.16	3.36	29.6	95.2
Mean Transfer Size (bytes)	12,993	13,993	5,918	21,188	9,838	12,942
Median Transfer Size (bytes)	2,503	2,674	1,898	4,179	4,542	3,849
Mean File Size (bytes)	33,879	41,511	16,166	32,500	13,497	41,337
Median File Size (bytes)	5,313	2,994	1,442	5,943	1,994	3,044

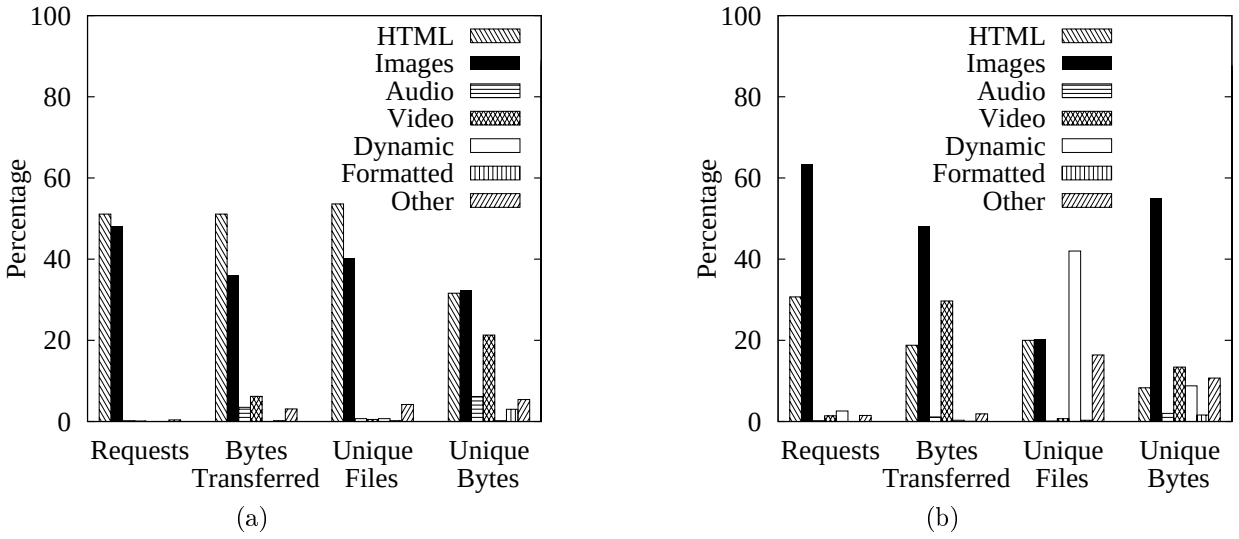


Figure 1: Distribution of Document Types: (a) NCSA Server; (b) NASA Server

4 Workload Characterization

This section presents a detailed analysis of file referencing behaviours on Internet Web servers, as well as a look at file sizes, transfer sizes, and the effect of user aborts in Web server workloads. We begin with an analysis of file and transfer size distributions.

4.1 File and Transfer Size Distributions

Figure 2 shows the cumulative distribution of the sizes of the distinct documents (files) transferred by each site. While there are a few very small files (< 100 bytes) at each of the sites, most files appear to be in the range of 100 - 100,000 bytes, with less than 10% larger than 100,000 bytes. This distribution is consistent with the file size distribution reported by Braun and Claffy [10].

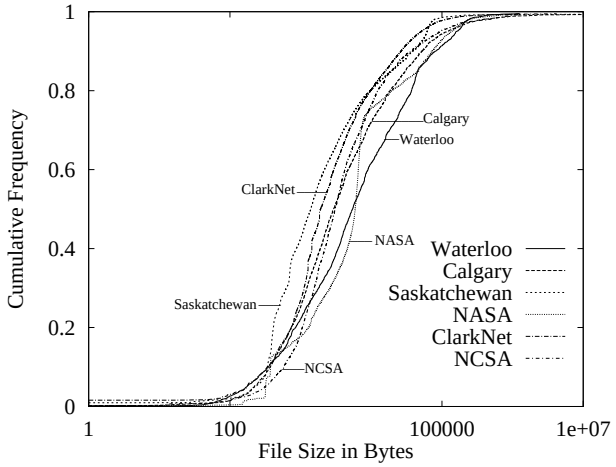


Figure 2: Distribution of File Sizes, by Server

A more rigourous study shows that the observed file size distributions are *heavy-tailed* (i.e., the tail of the distribution matches well with the Pareto distribution [19, 29], for $\alpha \approx 1$). In particular, the tails of the distributions (for files larger than 10,000 bytes) fit well to a Pareto distribution with $0.93 \leq \alpha \leq 1.33$. A similar analysis finds that the transfer size distributions are also heavy-tailed, although not as heavy-tailed as the file size distributions (for transfers of 10,000 bytes or more, estimates of α are in the range $1.28 \leq \alpha \leq 2.07$). Similar observations have been noted in the literature [11, 28, 29], and are confirmed in all six of our data sets. This heavy-tailed characteristic for the file and transfer size distributions is thus added to Table 1.

Figure 3(a) illustrates this behaviour for the file size distribution in the ClarkNet data set, using log-log comple-

Table 5: Statistics on Distinct Documents for All Data Sets

Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Distinct Requests/Total Requests	2.5%	1.5%	0.9%	0.3%	1.1%	1.0%
Distinct Bytes/Total Bytes	6.0%	3.5%	2.0%	0.3%	1.5%	2.4%
Distinct Files Accessed Only Once	32.4%	22.6%	42.0%	42.1%	32.4%	35.0%
Distinct Bytes Accessed Only Once	34.7%	19.8%	39.1%	15.3%	24.8%	39.1%

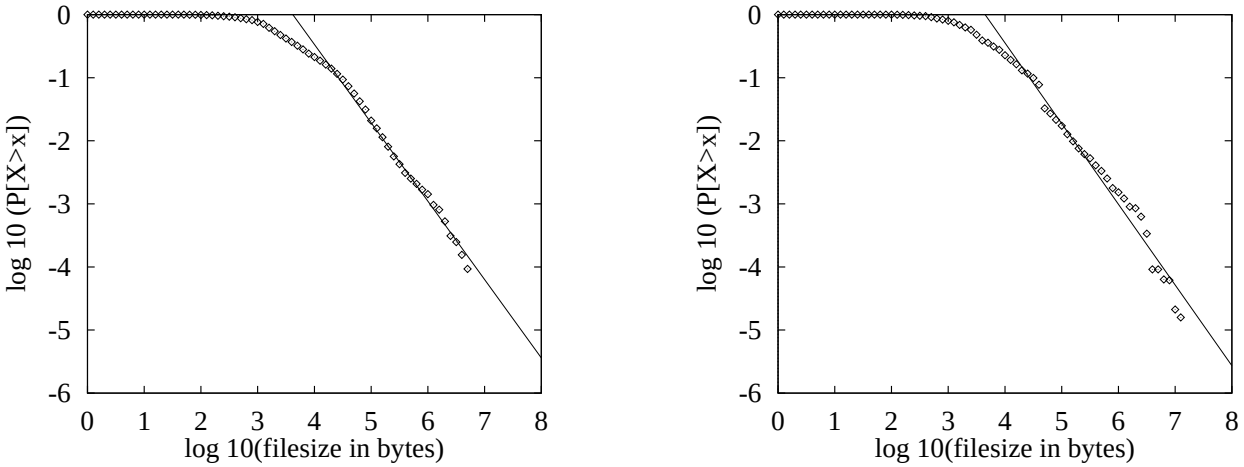


Figure 3: Log-Log Complementary Distribution Plots to Illustrate Heavy Tails: (a) File Size Distribution (ClarkNet Server, $\alpha = 1.24$); (b) Transfer Size Distribution (Calgary Server, $\alpha = 1.28$)

mentary distribution plots, and linear regression to estimate α. Figure 3(b) shows the heavy-tail property in the transfer size distribution for the Calgary data set. Table 6 summarizes the α values determined for all six data sets.

4.2 File Referencing Behaviour

This subsection looks at a number of different characteristics in the file referencing patterns at Internet Web servers. The analysis focuses on concentration of references, temporal locality, inter-reference times, and geographic distribution of references.

4.2.1 Concentration of References

Our first analysis focuses on the frequency of reference for different Web documents. Clearly, not all Web documents are created equal. Some are extremely "hot" and popular documents, accessed frequently and at short intervals by many clients at many sites. Other documents are accessed rarely, if at all.

We illustrate this non-uniform referencing pattern, which we call *concentration* [35], by sorting the list of distinct files into decreasing order based on how many times they were accessed, and then plotting the cumulative frequency of requests versus the fraction of the total files referenced. The resulting plot for all six data sets is shown in Figure 4.

Figure 4(a) illustrates the non-uniform pattern of file referencing behaviour: 10% of the distinct documents were responsible for 80-95% of all requests received by the server, at each of the six sites. Similar results (not shown here) are observed for the bytes transferred by the server [4]. Among the six data sets, the NCSA data set shows the most concentration, while the Calgary data set shows the least. A typical plot of document reference count versus activity rank is shown in Figure 4(b).

This concentration phenomenon is another common characteristic in our Web server logs, and is thus added to Table 1. Braun and Claffy have reported similar results for NCSA's Web server in an earlier study [10].

4.2.2 Temporal Locality

Access logs were next analyzed to look for temporal locality in the file referencing behaviour. Temporal locality refers to the notion of the same document being re-referenced frequently within short intervals. Note that the temporal locality property is orthogonal to the concentration property just analyzed: concentration refers to the aggregate reference counts for documents (regardless of the referencing order), while temporal locality refers to the relative order in which documents are referenced (regardless of their reference counts).

Temporal locality can be measured using the standard LRU (Least Recently Used) stack-depth analysis. When a document is initially referenced, it is placed on top of the LRU stack (i.e., position 1), pushing other documents down in the stack by one location. When the document is subsequently referenced, its current location in the LRU stack is recorded, and then the document is moved back to the top of the stack, pushing other documents down, as necessary. When the entire log has been processed in this fashion, temporal locality in referencing behaviour is

Table 6: Estimates of α for File and Transfer Size Distributions (All Data Sets)

Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
$\hat{\alpha}$ for File Size Distribution	1.33	0.99	1.07	1.19	1.24	0.93
$\hat{\alpha}$ for Transfer Size Distribution	1.74	1.28	1.48	2.07	1.62	1.42

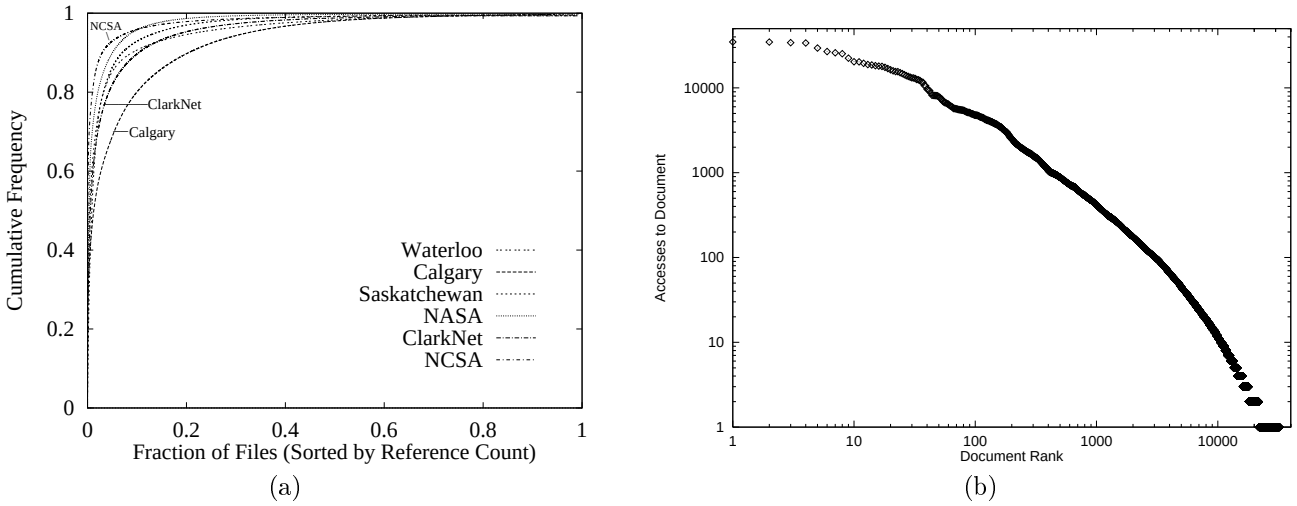


Figure 4: Concentration of References: (a) Cumulative Distribution (All Servers); (b) Reference Count versus Rank (ClarkNet Server)

manifested by a high probability of references to locations at or near the top of the LRU stack.

Figure 5(a) shows the results of our LRU stack-depth analysis for all six data sets. The Calgary data set shows the highest degree of temporal locality, while the ClarkNet data set shows the least. Overall, the degree of temporal locality observed is much lower than that observed in file systems [35]. Our speculation is that client-side caching mechanisms remove temporal locality from the reference stream seen at the server, as has been shown in other client-server environments [15].

Note, however, that the degree of temporal locality observed depends on the granularity at which the LRU stack frequencies are classified. For example, Figure 5(b) shows a coarser granularity analysis of temporal locality for the Saskatchewan server. The leftmost histogram shows LRU stack depth frequencies for the first 200 positions (computed using 20 intervals each of width 10), and the rightmost histogram shows the corresponding LRU stack depth frequencies for the first 2000 positions (computed using 10 intervals each of width 200). In both histograms, the rightmost spike represents the cumulative reference frequencies for all remaining stack positions. Clearly, a greater degree of temporal locality is seen at coarser levels of granularity. The measurement results in Figure 5(b) are consistent with those reported in [3].

4.2.3 Inter-Reference Times

Our next analysis focuses on the inter-reference times for documents. The purpose of the analysis is to determine if the request arrival process is Poisson (i.e., the time between successive requests are exponentially distributed and

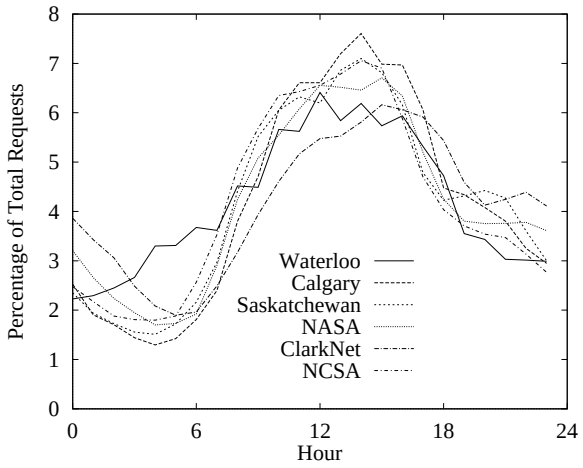
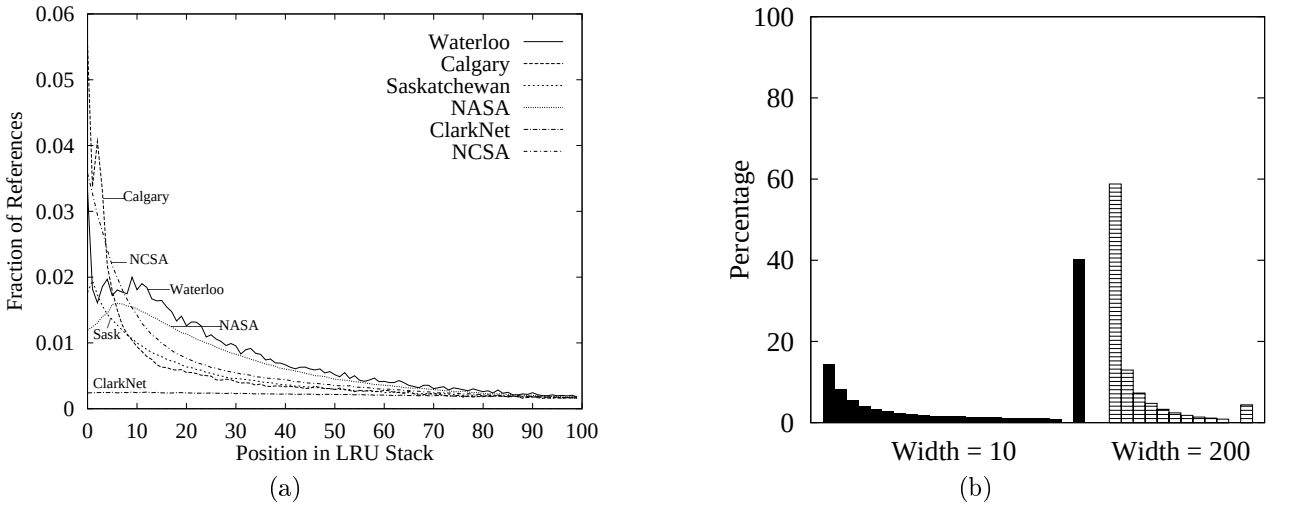


Figure 6: Distribution of Hourly Request Arrival Rates

independent).

For our traces, the aggregate reference stream is definitely not Poisson, since time-of-day effects produce non-stationary request rates (see Figure 6). Thus it is not possible to model the inter-reference times with a simple, fixed-rate Poisson process (this is similar to the problem faced by Paxson and Floyd for telnet and ftp session arrivals [29]).

The next simplest model, as suggested by Paxson and Floyd [29], is to postulate that the inter-reference times during fixed interval lengths can be well modeled by homogeneous Poisson processes. Using this approach, the inter-reference times were again studied, this time over one hour intervals. Under these conditions, the aggregate request stream still does not appear to be a Poisson process. Fig-



(a) (b)

Figure 5: Temporal Locality Characteristics: (a) Fine-Grain Results for All Six Data Sets; (b) Coarse-Grain Results for the Saskatchewan Data Set

ure 7(a) shows a comparison of the actual inter-reference time distribution (for a one hour interval from the NASA trace, with 1166 requests processed within this period, for a mean inter-arrival time of 3.09 seconds) with an exponential distribution using the same mean arrival rate. While the empirical distribution follows the exponential distribution quite closely for short time scales, the empirical distribution has a much heavier tail (similar observations are made by Mogul [22]). Furthermore, inter-reference times show positive correlation (see Figure 7(b)), perhaps because of machine-initiated requests [29].

There is evidence, however, that the request arrival process for *individual* documents *is* Poisson. For this analysis, we define a *busy document* as a document that is referenced at least 50 times in a one hour interval, in at least 25 different (non-overlapping, but not necessarily contiguous) one-hour intervals in the trace. There were 135 such documents in our traces: 0 from the Waterloo trace, 0 from Calgary, 2 from Saskatchewan, 34 from NASA, 44 from ClarkNet, and 55 from NCSA.

Figure 8(a) shows the results of the analysis of the inter-reference times for these busy documents, using the tests described in [29]. There is one data point plotted for each document analyzed. If the reference stream is truly Poisson, we would expect 95% of the tested intervals to pass the test for exponentially distributed inter-arrivals and uncorrelated inter-arrivals (i.e., they would appear in the upper right hand corner of the graph). Figure 8(a) shows that most of the files tested show reference behaviour consistent with Poisson arrivals. Furthermore, if the analysis is tightened to exclude one-hour intervals where more than 2.5% of the reported inter-arrival times are zero[4] then the files cluster even more closely in the upper right hand corner[5] (see Figure 8(b)). Thus, the request streams for individual busy documents appear to be Poisson.

4.2.4 Geographic Distribution

Our final analysis of file referencing behaviour examines the geographic distribution of document requests. This analysis makes use of the IP addresses of the requesting hosts in the access log. In particular, the network component of the IP address (based on a Class A, Class B, or Class C address) is used to determine if a requesting host is *local* or *remote* relative to the Web server. The network identifier in each IP address is further used to identify the number of remote networks that access the Web server.

Table 7 shows the geographic distribution of requests and bytes transferred at the six sites. For example, 76.4% of all the requests to the Waterloo server came from remote hosts, while local hosts generated the remaining 23.6% of the requests. In terms of bytes transferred, 80.7% of the requested bytes were transferred to remote hosts, with 19.3% to local hosts. The rest of the table is organized similarly.

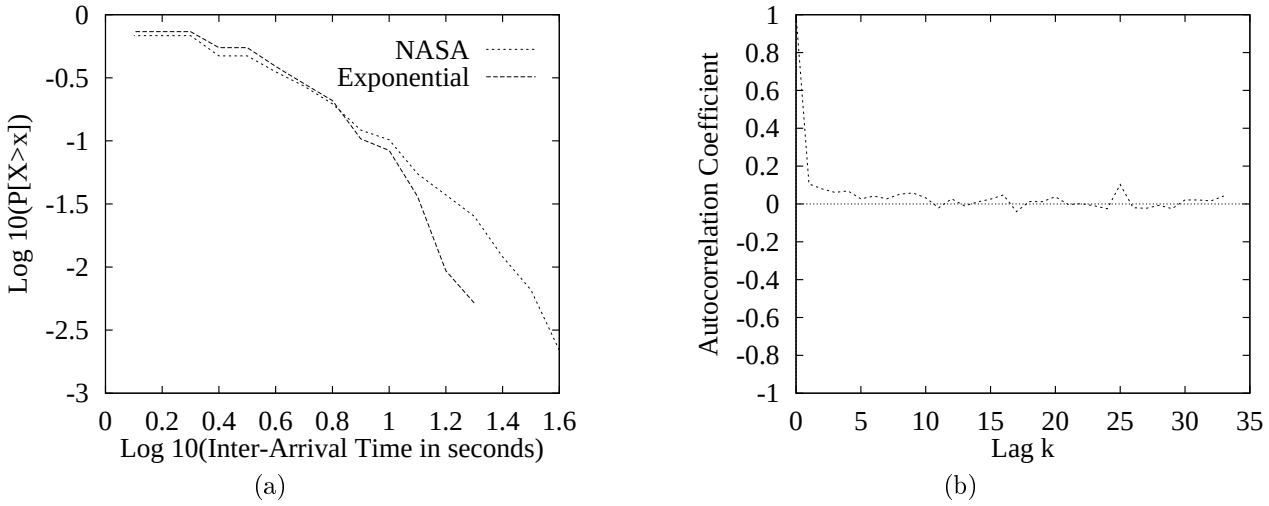
On all six Web servers, remote hosts send the most requests and receive the most data. Remote hosts account for over 75% of requests on all but one server (Calgary), and well over half of the total bytes transferred on all servers. This observation is reported in Table 1 as another common workload characteristic.[6]

The local access patterns at the Saskatchewan and Waterloo servers are quite similar. The similarity is likely caused by the use of the Web in teaching and research activities. The access pattern at NCSA, NASA, and ClarkNet is substantially different, with remote accesses accounting for almost all of the requests and transferred data. The likely explanation for this behaviour is the different client bases

[4]The server logs have a relatively coarse (1 second) timestamp resolution, and the presence of zeros makes the testing for exponentiality difficult.

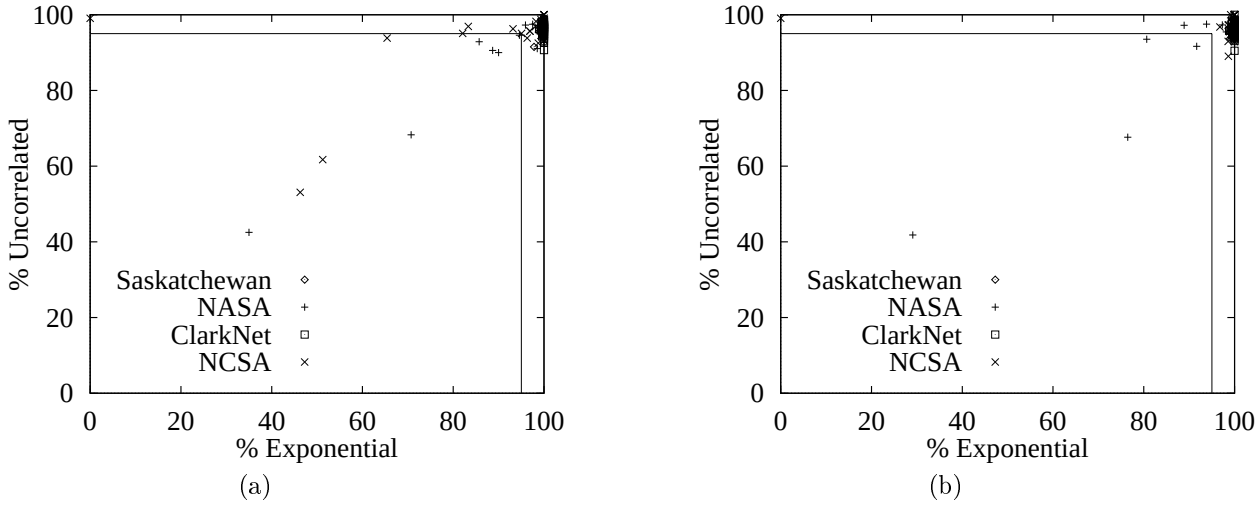
[5]There are several outliers in this plot. The outlier at (0,100) is a file called "InformationServers/.Test/TestPattern.html", which was requested every 30 seconds by a single host within the NCSA domain. The other two significant outliers were from the NASA data set. The first file was a live video feed of a shuttle launch, while the second was a dynamic file that performed a countdown to the launch.

[6]Clearly, this observation does not apply to Web servers inside firewalls, or on *intranets*.



(a) (b)

Figure 7: Inter-Reference Time Analysis (NASA Server): (a) Comparison to Exponential Distribution; (b) Auto-Correlation Function



(a) (b)

Figure 8: Inter-Reference Time Analysis (NASA Server): (a) Initial Analysis; (b) Refined Analysis

and service roles for these organizations (e.g., research, public relations, commercial Internet service provider).

Figure 9 shows the breakdown of the client requests by network address type (Figure 9(a)) and by continent of origin (Figure 9(b)). Approximately 80% of the requesting hosts were from North America, and approximately 60% had Class B IP addresses. These characteristics were practically identical in the five data sets that were analyzed.[7] Furthermore, across the five data sets, 10% of the networks accounted for at least 75% of the requests (Characteristic 10 in Table 1). Coupled with our earlier workload observations, the latter two workload characteristics (Characteristic 9 and Characteristic 10) clearly suggest that geographic caching of Web documents could be highly effective.

[7] The Calgary data set was not included in this analysis since the "sanitized" logs that we received from the University of Calgary did not show host names or IP addresses for each request, but only a boolean indicator of LOCAL or REMOTE.

4.3 Aborted Connections

Several Web documents appeared in an access log multiple times, with the same URL each time, but with different transfer sizes at different points in the log. There are two possible causes for these "anomalies" in the Web server access logs. First, a user may edit and physically change a Web document at any time. Second, a Web client may abort a connection in the midst of a document transfer (i.e., the user clicks on the "Stop" button in the Web browser).

An analysis was thus performed on these events to assess the impact of user aborts and file modifications on the results reported in this paper. For example, Figure 10(a) illustrates the number of aborts and file modifications detected per week in the Saskatchewan data set, compared to the number of requests per week. The overall abort rate is approximately 1.4%, while approximately 1.0% of requests are for files that have been modified.

Figure 10(b) analyzes aborts and file modifications at the Waterloo server from the standpoint of whether the

Table 7: Geographic Distribution of Requests for All Data Sets

	Remote Hosts					
Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
% All Requests	76.4	53.6	75.1	93.7	98.1	98.8
% All Bytes	80.7	63.4	75.2	97.3	98.4	99.5

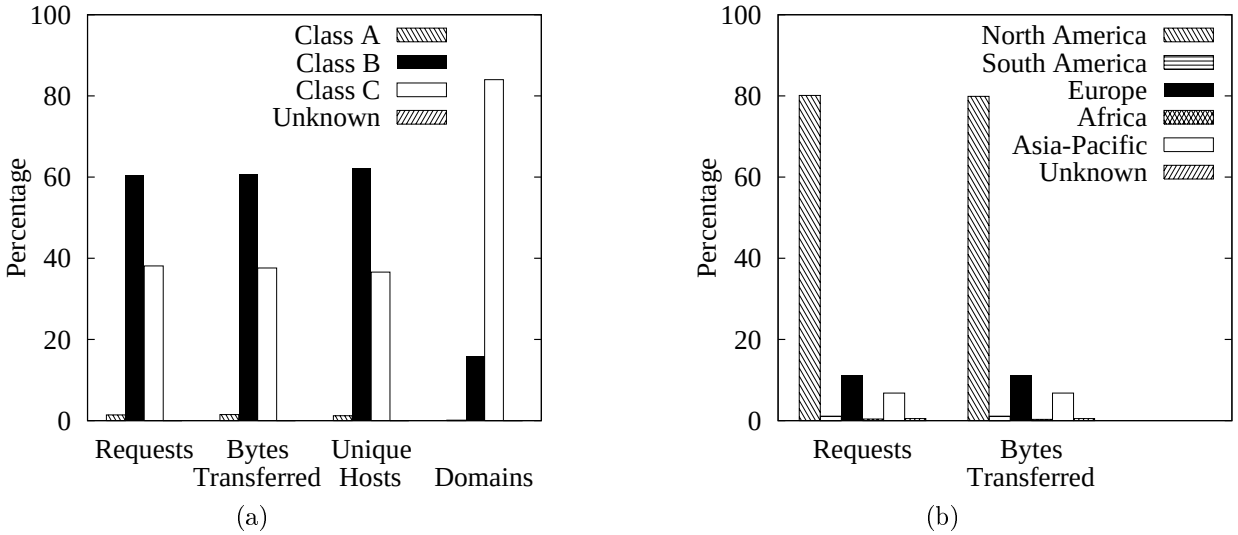


Figure 9: Analysis of Client IP Addresses: (a) Breakdown by Network Class (NCSA Server); Geographic Distribution of Requests by Continent (ClarkNet Server)

requesting client is local or remote. In this data set, 76% of the incoming requests are from remote clients, and 98% of the aborted connections (which constitute 4.8% of the total requests) are aborted by remote clients. As expected, remote users are more likely to abort a connection than are local users, since network bandwidth to the Web server is often lower for remote users than local users. Regarding file modifications, 81% of the time, the first request to a modified file is made by a local client. This is likely the result of a user visually inspecting the changes that were just made to that file (or Web page).

Table 8 summarizes information about aborted connections and file modifications for all six data sets. While the number of aborted connections is quite low, the number of bytes transferred by aborted connections is larger.

4.4 Summary

This section has presented a detailed study of Web server workload characteristics. Results were presented for file size distributions, file referencing patterns, and aborted connections in Web server workloads.

From the analyses reported in this section, five additional workload characteristics have been identified. These characteristics appear in the last five rows of Table 1.

5 Performance Implications

We conclude our paper with a discussion of caching and performance issues for Internet Web servers. Despite the low temporal locality seen in most Web server workloads, caching still appears to be a promising approach to improving Web performance because of the large number of requests for a small number of documents (Characteristic 4 from Table 1), the concentration of references within these documents (Characteristic 7), and the small average size of these documents (Characteristic 3). We intentionally leave unspecified the location[8] of the cache, focusing instead on the use of our workload characteristics to estimate the maximum performance improvement possible with Web server caching. For simplicity, the discussion assumes that all Web documents are read-only (i.e., never modified), and that file-level (not block-level) caching is used. Misses due to "cold start" are also ignored.

5.1 A Tradeoff: Requests versus Bytes

There are two main elements that affect the performance of a Web server: the number of requests that a server must process, and the number of data bytes that the server must transfer (i.e., disk I/O's, packets).

[8]Several logical choices exist: (1) at the client, or the client's network, to reduce requests to a remote server; (2) at the server, or the server's network, to reduce disk accesses and/or byte transfers on the server's network; (3) in the network itself, to reduce repeated "pulls" of the same document across a geographic region of the network; and (4) a combination of the above.

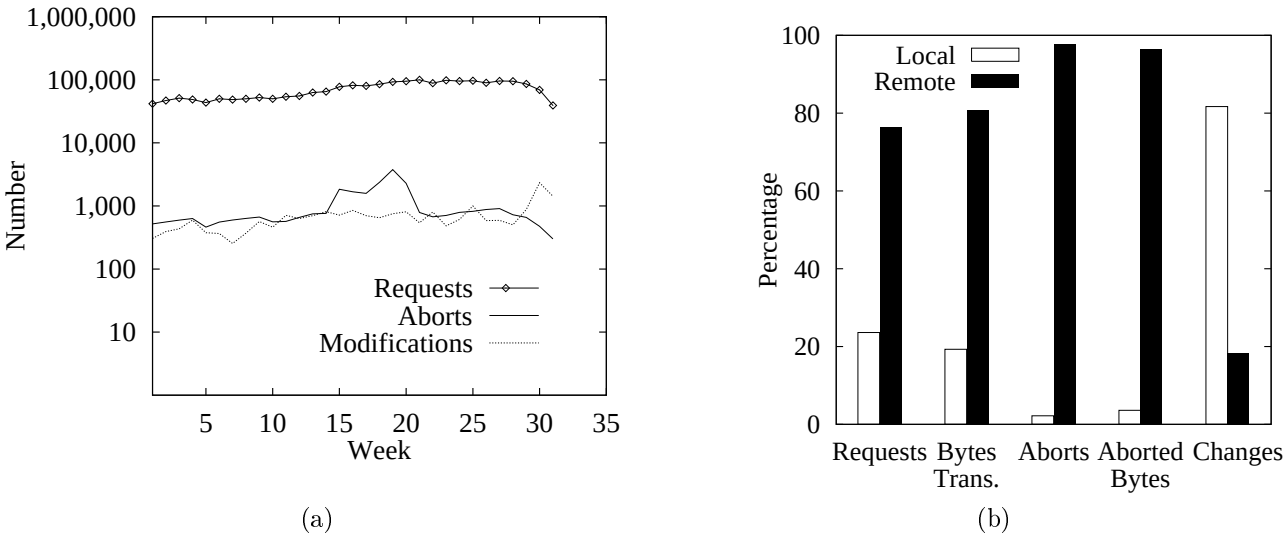


Figure 10: Analysis of Aborted Connections: (a) Aborted Connections and File Modifications (Saskatchewan Server); (b) Breakdown of Requests by Local and Remote (Waterloo Server)

Table 8: Aborted Connections and File Modifications

Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
% of Connections Aborted	4.8	2.4	1.4	0.5	0.005	2.0
% of Bytes Aborted	10.3	7.8	5.6	2.2	0.006	8.4
% of Requests for Changed Files	2.5	15.3	1.0	8.6	0.8	0.2

There is thus a choice to be made between caching designs that reduce the number of requests presented to Internet Web servers, and caching designs that reduce the volume of network traffic[9]. Both approaches represent possible avenues for improving Web server performance, but optimizing one criterion does not necessarily optimize the other. The choice between the two depends on which resource is the bottleneck: CPU cycles at the server, or network bandwidth.

We illustrate this tradeoff in Figure 11. While the discussion here focuses only on the ClarkNet data set, similar observations apply for the other data sets.

The topmost graph (Figure 11(a)) illustrates the relationship between the size of files on a Web server (from Figure 2), the number of requests to those files, and the number of data bytes that requests to those files generate (i.e., the "weighted value" obtained from the product of file size and number of times that a file is requested). This graph shows that 80% of all the documents requested from the ClarkNet server were less than 10,000 bytes in size. 76% of all requests to the server were for files in this category. Thus, caching a large number of small files would allow the server to handle most of the requests in a very efficient manner. However, Figure 11 also points out that the requests to files less than 10,000 bytes in size generate only 26% of the data bytes transferred by the server. Furthermore, looking at the tail of the distribution, documents over 100,000 bytes in size are responsible for 11% of the bytes transferred by the server, even though less than 0.5% of the requests are for files in this category (Characteristic 6). What this means is that in order to reduce the number of bytes transferred by the server as much as possible, a few large(r) files would have to be cached. That is, the server must sacrifice on "cache hits" for many small requests to save on bytes transferred for large requests.

The remaining two plots in Figure 11 illustrate the tradeoff in more detail. The middle plot (Figure 11(b)) shows the results for a cache designed to maximize cache hits for requests (i.e., to reduce the number of requests to the server). In this graph, the top line represents the cache hit rate for requests, the bottom line represents the cache size, and the middle line represents the potential savings in bytes transferred by the server when the cache is present. In this design, for example, caching 10% of the server's distinct files (namely, the most frequently accessed documents) for the ClarkNet data set results in a cache hit rate of 90% (the top line in the graph). The documents in the cache, which represent the potential savings in bytes transferred, account for 84% (the middle line in the graph) of the bytes transferred by the server, and the cache size would need to hold 8.3% (the bottom line in the graph) of the total distinct bytes requested in the server access log.

The bottom plot (Figure 11(c)) represents the results for a cache designed to reduce bytes transferred. In this graph, the top line represents the savings in bytes transferred, the bottom line represents the cache size, and the middle line represents the cache hit rate. In this design, for example, caching 10% of the server's files (namely, the 10% of the documents that account for the most bytes transferred)

[9]Clearly, reducing the number of requests also reduces the volume of network traffic, but the main focus of the two approaches is different, as will be shown.

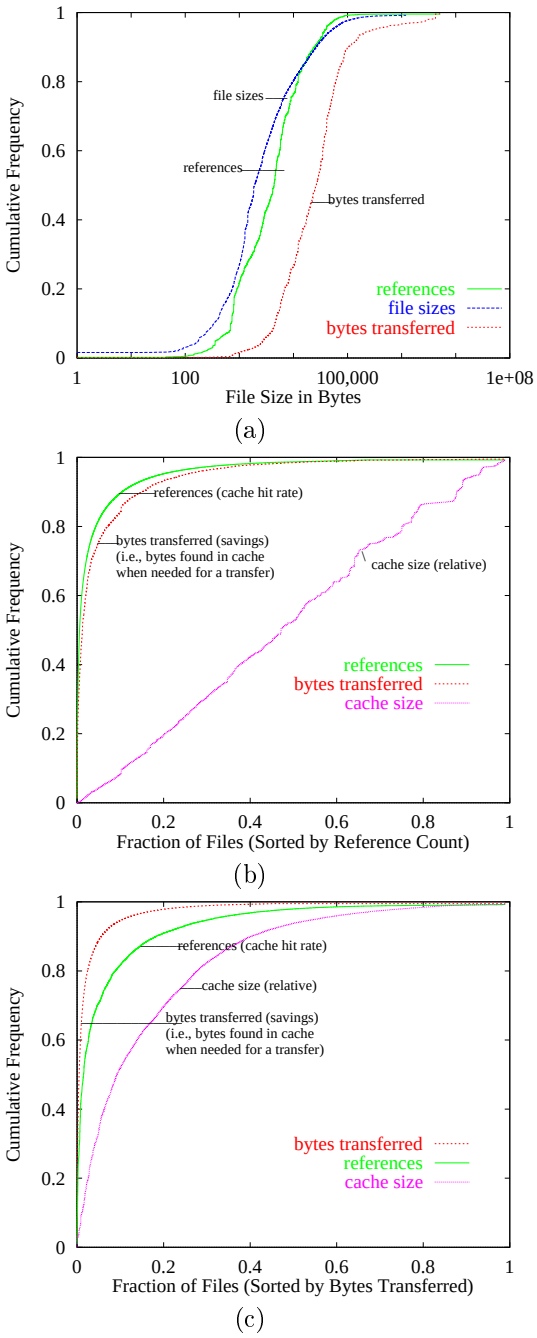


Figure 11: Comparison of Caching and Performance Issues for ClarkNet Data

results in an 82% cache hit rate (the middle line). The documents in the cache would account for 95% (the top line) of the bytes transferred, but the cache would have to be large enough to contain 52% (the bottom line) of the distinct bytes represented in the server access log.

Table 9 presents the results for each of the six data sets. The top half of Table 9 gives the results for a cache designed to maximize the hit rate. Across all six data sets, cache hit rates of 80-96% are possible using relatively small cache sizes (6-45 MB, representing 5-20% of the unique bytes). This cache design also saves 54-91% of the bytes transferred.

The bottom half of Table 9 shows the performance results for a cache designed to maximize the byte savings. This study found that savings of 90-97% in bytes transferred are possible, as well as cache hit rates of 61-92%. However, larger cache sizes (48-478 MB, representing 44-72% of the unique bytes) are needed to achieve these results. Clearly there is a tradeoff to be made in cache size, cache hit rate, and bytes transferred by the server.

5.2 Cache Management Issues

Our final comments concern "one timers", cache replacement strategies, and thresholding approaches to cache management. We have investigated these caching issues using trace-driven simulations based on our Web server workloads [4, 6].

First, the "one time" referencing (Characteristic 5) of Web documents is a concern.[10] While this effect could be simply an artifact of the finite duration of the access logs studied, or something as innocent as the deletion or renaming of Web documents, the effect is present across *all* access log time durations studied. This one-time referencing behaviour means that, depending on the cache replacement policy, up to one-third of a server cache, on average, could be cluttered with useless files. Techniques to expunge such files from a cache are desirable.

Second, the fact that strong temporal locality was *not* present in all data sets suggests that LRU as a cache replacement policy may not work well for *all* servers. Policies such as Least Frequently Used (LFU) may be more attractive because of the concentration of references (Characteristic 7), and also because LFU easily deals with one-timers.

Trace-driven simulations do indeed show that LFU is consistently superior to LRU, and indeed superior to the size-based replacement policy advocated in [36]. For example, Figure 12 presents a comparison of frequency-based (namely, LFU*-Aging, as described in [6]), recency-based (LRU), and size-based (namely, the $\lfloor log_2(SIZE) \rfloor$ policy proposed in [36]) replacement policies for the ClarkNet data set, using the two standard metrics of Hit Rate and Byte Hit Rate for documents in the cache [36]. As shown in Figure 12, the frequency-based policy provides consistently superior performance for both metrics. Note that the size-based policy, which provides the best hit rate in [36], can actually provide a significantly worse byte hit rate than other policies. Similar observations apply across all six of our data sets. Adding caching *partitions* based on document types helps in some cases, but does not change the relative ordering of the policies studied [4].

Third, there may be merit in using "size thresholds" in cache management, to better cope with the "heavy tailed" Pareto distribution of file sizes (Characteristic 6), and the issues raised in Section 5.1. For example, two such threshold policies might be "never cache a document larger than X bytes" (because it uses up too much cache space, and

[10] The advent of *Web crawlers* may change this characteristic to be "N timers", for some small integer N. However, the argument that we make here still applies.

Table 9: Cache Performance for All Data Sets (when caching 10% of files)

Maximizing Hit Rate						
Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Cache Hit Rate (%)	91	80	93	96	90	96
Byte Savings (%)	75	54	86	91	84	90
% of Unique Bytes	5.5	5.3	5.1	19.5	8.3	6.7
Cache Size (MB)	6	14	13	40	34	45
Maximizing Byte Savings						
Item	Waterloo	Calgary	Saskatchewan	NASA	ClarkNet	NCSA
Cache Hit Rate (%)	73	61	86	88	82	92
Byte Savings (%)	91	90	96	97	95	97
% of Unique Bytes	46.2	64.4	44.4	44.5	52.0	71.7
Cache Size (MB)	48	170	111	91	216	478

adversely impacts hit rate), or "never cache a document smaller than Y bytes" (because it does not save much on bytes transferred by the server). Trace-driven simulation experiments confirm this intuition. Figure 13 shows the simulation results for the NCSA data set using four possible thresholding policies: upper threshold only (100 kilobytes), lower threshold only (500 bytes), lower and upper threshold (500 bytes and 100 kilobytes, respectively), and no threshold. An upper threshold only provides a slight improvement in cache hit rate, but a substantial decrease in performance for byte hit rate. A lower threshold policy, on the other hand, provides the same byte hit rate, but with a lower cache hit rate. Overall, the performance advantages of size-based thresholding policies are negligible.

6 Conclusions

This paper has presented a detailed workload characterization study for Internet World Wide Web servers. The study used logs of Web server accesses at six different sites: three from university environments, two from scientific research organizations, and one from a commercial Internet provider. The logs represent three different orders of magnitude in server activity, and span two different orders of magnitude in time duration.

From these logs, we have been able to identify ten common characteristics in Web server workloads. These characteristics appear in Table 1 at the start of the paper.

The observed workload characteristics were used to identify two possible strategies for the design of a caching system to improve Web server performance, and to determine bounds on the performance improvement possible with each strategy. The performance study identified the distinct tradeoff between caching designs that reduce network traffic, and caching designs that reduce the number of requests presented to Internet Web servers. While the two approaches are somewhat at odds with each other, both represent possible avenues for improving Web server performance. Our results show that caching to reduce the number of requests may be more effective than caching to reduce bytes transferred. The observed workload characteristics also suggest cache management strategies, such as frequency-based replacement, that can improve the cache hit rate and byte hit rate for Internet Web servers.

Several relatively recent Web forces may someday undermine or change our ten Web server workload characteristics. These forces include: Web crawlers, which could reduce the one-time referencing phenomenon; improved protocols for Web interaction, which could improve bandwidth, encouraging users to request larger files; small-scale and large-scale Web caching architectures, which could alter the request streams that servers must handle; and a growing trend toward the use of video, audio, and interactivity on the Web (e.g., CGI, Java), which could change the distributions of document types, file sizes, and transfer sizes, as well as the median transfer size. Our workload characterization effort provides an important baseline from which to evaluate the impact of these forces on future Web server workloads.

Acknowledgements

The authors are grateful to the following people for making their Web server access logs available for our study: Jamie Hodge, Department of Computer Science, University of Waterloo; Robert Fridman, Department of Computer Science, University of Calgary; Earl Fogel, Department of Computing Services, University of Saskatchewan; Jim Dumoulin, NASA (Kennedy Space Center); Stephen Balbach, ClarkNet; and Robert McGrath, NCSA.

Funding for this research was provided by NSERC Research Grant OGP0120969, and by an NSERC Postgraduate Scholarship. Part of this work utilized the Mass Storage System at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.

This research was motivated by a discussion with Vern Paxson in May 1995. Vern Paxson was also instrumental in establishing the Internet Traffic Archive.

References

[1] M. Abrams, C. Standridge, G. Abdulla, S. Williams,

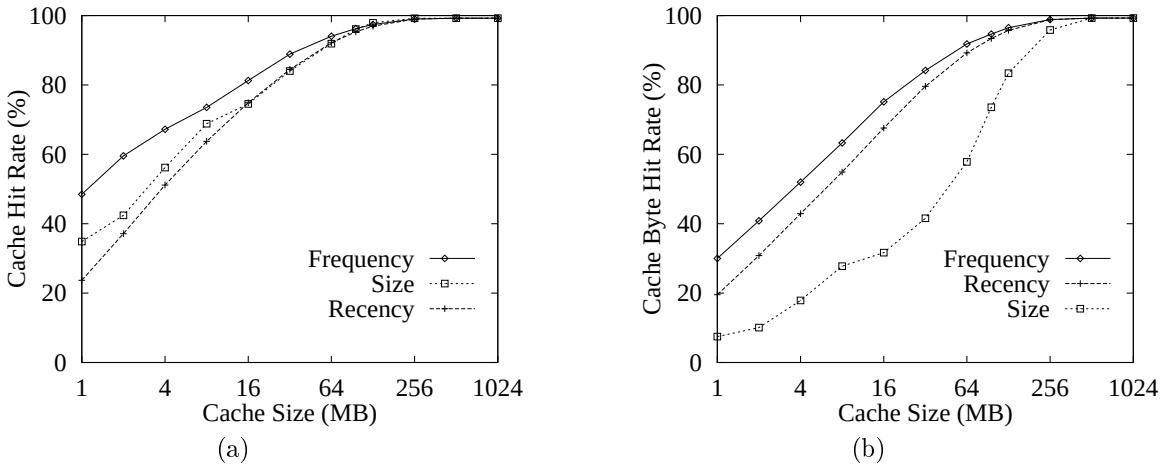


Figure 12: Comparison of Frequency-based, Size-based, and Recency-based Replacement Policies for the ClarkNet Data Set: (a) Document Hit Rate; (b) Byte Hit Rate

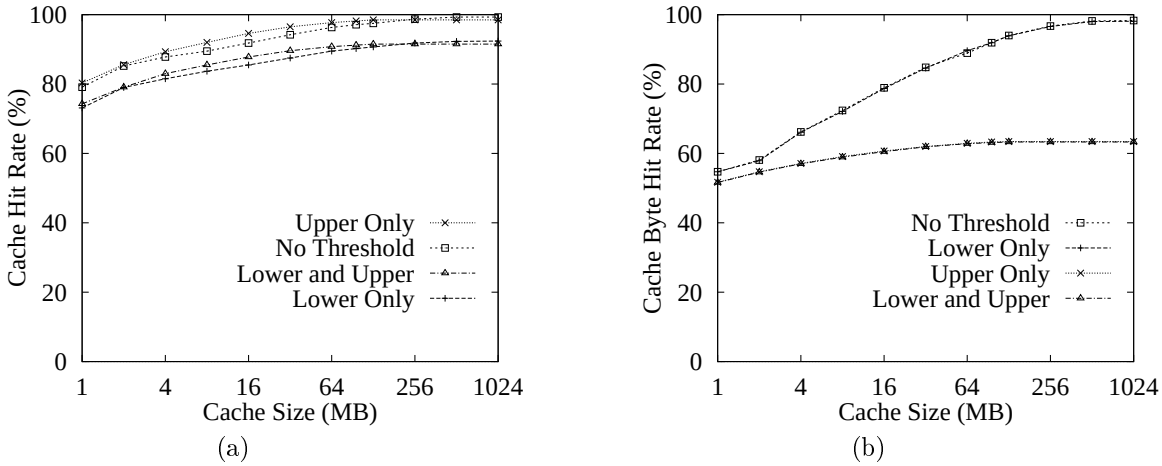


Figure 13: Comparison of Size-based Thresholding Policies for the NCSA Data Set: (a) Document Hit Rate; (b) Byte Hit Rate

and E. Fox, "Caching Proxies: Limitations and Potentials", *Electronic Proceedings of the Fourth World Wide Web Conference '95: The Web Revolution*, Boston, MA, December 11-14, 1995.

[2] J. Almeida, V. Almeida and D. Yates, "Measuring the Behavior of a World-Wide Web Server", *Seventh Conference on High Performance Networking (HPN)*, White Plains, NY, pp. 57-72, April 1997.

[3] V. Almeida, A. Bestavros, M. Crovella, and A. de Oliveira, "Characterizing Reference Locality in the WWW", *Proceedings of the Fourth International Conference on Parallel and Distributed Information Systems (PDIS '96)*, December 1996.

[4] M. Arlitt, "A Performance Study of Internet Web Servers", M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, June 1996.

[5] M. Arlitt and C. Williamson, "Web Server Workload Characterization: The Search for Invariants", *Proceedings of the 1996 ACM SIGMETRICS Conference*, Philadelphia, PA, pp. 126-137, May 1996.

[6] M. Arlitt and C. Williamson, "Trace-Driven Simulation of Document Caching Strategies for Internet Web Servers", *Simulation Journal*, Vol. 68, No. 1, pp. 23-33, January 1997.

[7] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Nielsen and A. Secret, "The World-Wide Web", *Communications of the ACM*, 37(8), pp. 76-82, August 1993.

[8] A. Bestavros, R. Carter, M. Crovella, C. Cunha, A. Heddaya and S. Mirdad, "Application-Level Document Caching in the Internet", *Proceedings of the Second International Workshop on Services in Distributed and Networked Environments (SDNE '95)*, Whistler, BC, Canada, pp. 166-173, June 1995.

[9] J. Bolot and P. Hoschka, "Performance Engineering of the World-Wide Web: Application to Dimensioning and Cache Design", *Electronic Proceedings of the Fifth International World-Wide Web Conference*, Paris, France, May 6-10 1996.

[10] H. Braun and K. Claffy, "Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server", *Electronic Proceedings of the Second World Wide Web Conference '94:*

Mosaic and the Web, Chicago, Illinois, October 1994.

[11] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", *Proceedings of the 1996 ACM SIGMETRICS Conference*, Philadelphia, PA, pp. 160-169, May 1996.

[12] C. Cunha, A. Bestavros and M. Crovella, "Characteristics of WWW Client-Based Traces", Technical Report BU-CS-95-010, Boston University Computer Science Department, 1995.

[13] P. Danzig, M. Schwartz and R. Hall, "A Case for Caching File Objects Inside Internetworks", *Proceedings of ACM SIGCOMM '93*, San Francisco, California, pp. 239-248, September 1993.

[14] P. Danzig, S. Jamin, R. Cáceres, P. D. Mitzel, and D. Estrin, "An Empirical Workload Model for Driving Wide-Area TCP/IP Network Simulations", *Internetworking: Research and Experience*, 3(1), pp. 1-26, March 1992.

[15] K. Froese and R. Bunt, "The Effect of Client Caching on File Server Workloads", *Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences*, January 1996.

[16] S. Glassman, "A Caching Relay for the World Wide Web", *First International Conference on the World Wide Web*, Geneva, Switzerland, May 1994.

[17] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West, "Scale and Performance in a Distributed File System", *ACM Transactions on Computer Systems*, Vol. 6, No. 1, pp. 51-81, February 1988.

[18] Sun Microsystems, "Javasoft Home Page", 1996.

[19] N. Johnson and S. Kotz, Editors-in-Chief, *Encyclopedia of Statistical Sciences, Volumes 6 and 9*, John Wiley & Sons, Inc., New York, 1988.

[20] T. Kwan, R. McGrath, and D. Reed, "NCSA's World Wide Web Server: Design and Performance", *IEEE Computer*, Vol. 28, No. 11, pp. 68-74, November 1995.

[21] E. Markatos, "Main Memory Caching of Web Documents", *Electronic Proceedings of the Fifth World Wide Web Conference '96*, Paris, France, May 6-10, 1996.

[22] J. Mogul, "Network Behavior of a Busy Web Server and its Clients", Technical Report WRL-TR-95.5, Digital Western Research Laboratory, October 1995.

[23] J. Mogul, "The Case for Persistent-Connection HTTP", *Proceedings of ACM SIGCOMM '95*, Cambridge, MA, pp. 299-313, August 28 - September 1 1995.

[24] National Center for Supercomputing Applications, "NCSA httpd", 1994.

[25] Network Wizards, "Internet Domain Survey", 1996.

[26] NSFNET Statistics, April 1995.

[27] V. Padmanabhan and J. Mogul, "Improving HTTP Latency", *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*, Chicago, Illinois, October 1994.

[28] V. Paxson, "Empirically-Derived Analytic Models of Wide-Area TCP Connections", *IEEE/ACM Transactions on Networking*, Vol. 2, No. 4, pp. 316-336, August 1994.

[29] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", *Proceedings of ACM SIGCOMM '94* London, England, pp. 257-268, August 1994.

[30] V. Paxson, "Growth Trends in Wide Area TCP Connections", *IEEE Network*, Vol. 8, No. 4, pp. 8-17, July/August 1994.

[31] J. Sedayao, "Mosaic Will Kill My Network!", *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*, Chicago, Illinois, October 1994.

[32] M. Spasojevic, M. Bowman and A. Spector, "Using a Wide-Area File System Within the World-Wide Web", *Electronic Proceedings of the Second World Wide Web Conference '94: Mosaic and the Web*, Chicago, Illinois, October 1994.

[33] A. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, New Jersey, 1996.

[34] D. Wessels and K. Claffy, "Evolution of the NLANR Cache Hierarchy: Global Configuration Challenges", 1996.

[35] C. Williamson and R. Bunt, "Characterizing Short Term File Referencing Behaviour", Proceedings of the International Phoenix Conference on Computers and Communications (IPCCC), Phoenix, Arizona, pp. 651-660, March 1986.

[36] S. Williams, M. Abrams, C. Standridge, G. Abdulla and E. Fox, "Removal Policies in Network Caches for World-Wide Web Documents", *Proceedings of ACM SIGCOMM '96*, Stanford, CA, pp. 293-305, August 1996.

[37] World-Wide Web Frequently Asked Questions, April 11, 1996.

[38] N. Yeager and R. McGrath, *Web Server Technology: The Advanced Guide for World Wide Web Information Providers*, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1996.

Martin Arlitt

Martin Arlitt received his B.Sc. and M.Sc. from the University of Saskatchewan in 1994 and 1996, respectively. He is currently working as a Research Intern with the Broadband Information Systems Lab at Hewlett-Packard Laboratories in Palo Alto, California, where his interests include Internet Web servers, intranets, Web server performance evaluation, and Web server benchmarking. His email address is: `mfa126@cs.usask.ca`

Carey Williamson

Carey Williamson (M '91 / ACM '85) received his B.Sc.(Honours) from the University of Saskatchewan in 1985, and his Ph.D. from Stanford University in 1991. He is currently an Associate Professor in the Department of Computer Science at the University of Saskatchewan, in Saskatoon, Canada. His research interests include network traffic measurement, workload characterization, Web server performance evaluation, network simulation, high speed networking, and ATM. His email address is: `carey@cs.usask.ca`