Workload Characterization of the 1998 World Cup Web Site

Martin Arlitt, Tai Jin
Internet Systems and Applications Laboratory
HP Laboratories Palo Alto
HPL-1999-35(R.1)
September, 1999

E-mail: {arlitt, tai}@hpl.hp.com

World-Wide Web, workload characterization, performance, servers, caching, World Cup

This paper presents a detailed workload characterization study of the 1998 World Cup Web site. Measurements from this site were collected over a three month period. During this time the site received 1.35 billion requests, making this the largest Web workload analyzed to date. By examining this extremely busy site and through comparison with existing characterization studies we are able to determine how Web server workloads are evolving. We find that improvements in the caching architecture of the World-Wide Web are changing the workloads of Web servers, but that major improvements to that architecture are still necessary. In particular, we uncover evidence that a better consistency mechanism is required for World-Wide Web caches.

© Copyright Hewlett-Packard Company 1999

1 INTRODUCTION

The 16th Federation Internationale de Football Association (FIFA) World Cup was held in France from June 10th through July 12th, 1998. France '98, as the 16th FIFA World Cup was commonly called, was the most widely covered media event in history [36]. An estimated cumulative television audience of 40 billion watched the 64 matches of France '98, more than twice the cumulative television audience of the 1996 Summer Olympic Games in Atlanta. The Web site for France '98, `www.france98.com`, also proved to be very popular, receiving more than 1 billion client requests during the tournament.

This paper presents a detailed workload characterization of the France '98 Web site. This study provides insights on the current state of the World-Wide Web. By comparing the results of this study to earlier Web characterization studies we are able to determine how Web workloads are evolving as the Web increases in popularity and utilizes new technologies.

Some of the more significant characteristics that we observed in the World Cup workload, and the performance implications of these characteristics include:

- HTTP/1.1 clients are becoming more prevalent, accounting for 21% of all requests. Widespread deployment of HTTP/1.1 compliant clients and servers is necessary for the functionality of HTTP/1.1 to be fully utilized.
- 88% of all requests were for Image files; an additional 10% were for HTML files, indicating that most user interest was in static (i.e., cacheable) files.

- almost 19% of all responses were ‘Not Modified’, indicating that cache consistency traffic had a greater impact in the World Cup workload than in previous Web server workloads [3]
- the workload was quite bursty although over longer time scale (e.g., hours or more) the arrival of these bursts was quite predictable
- for timeouts of 100 seconds or less, many users sessions contained only a single request and a single response. We believe that this characteristic is due to the improved Web caching architecture that now exists. This characteristic has (possible) implications on both server and protocol design
- during periods of peak user interest in the World Cup site the volume of cache consistency traffic increased dramatically. This indicates that the lack of an efficient consistency mechanism (either the specification or utilization of one) is preventing Web caches from eliminating flash crowds in the network and at the servers, which is supposed to be one of the main benefits of Web caching.

Workload characterization plays an important role in systems design. It allows us to understand the current state of the system. By characterizing the system over time we can learn what effects changes to the system have had. Workload characterization is also crucial to the design of new system components. In this paper we focus on the characterization of a Web server workload. We compare our results to those from previous studies (e.g., [3][28]) to determine how Web server workloads have changed over time. Furthermore, the extremely heavy workload of the World Cup site allows us to predict what the workloads of future Web servers may look like, so that we may plan accordingly.

Web server workload characterization is only one of the necessary steps for understanding the changes occurring in Web traffic. Research efforts on Web client workloads (e.g., [5]), Web proxy workloads (e.g., [2][7][8][15][17][18][22][27][33]), network traffic characterizations (e.g., [34]) as well as HTTP analyses (e.g., [4][21][23][30]) are all required in order better understand the Web.

The remainder of the paper is organized as follows. Section 2 provides background information on the 1998 World Cup, focusing on the structure of the tournament. Section 3 introduces the World Cup Web site and describes the technology that it utilized. Section 4 discusses the collection and reduction of the data set used in the workload characterization study. Section 5 presents the results of our workload characterization study. Section 6 analyzes a particularly busy segment of the World Cup workload and compares the results to the overall study in Section 5. Section 7 describes in more detail the performance implications of the results from Section 5 and Section 6. Section 8 summarizes the contributions of our paper and lists areas of future work.

2 THE 1998 WORLD CUP

In order to better understand the nature of the workload from the France ’98 Web site knowledge of the tournament itself is required. In this section we provide a brief overview of the

World Cup, focusing on the France ’98 tournament in particular. Additional information on the World Cup and the France ’98 tournament is available on the FIFA Web site [20].

The FIFA World Cup is a tournament that is held once every four years to determine the best football (soccer) team in the world. This competition is open to all teams that represent the FIFA affiliated national football association of their respective countries. Due to the large number of teams interested in participating, a qualifying round is now used to select the teams that will play in the World Cup tournament. The qualifying round for France ’98 was held from March 1996 until November 1997. Of the 172 countries that entered the qualifying round 30 were selected to compete in France ’98, along with the host country, France, and the reigning champions, Brazil.

France ’98 began on June 10^{th}, 1998 and ended on July 12^{th}, 1998. The tournament consisted of several rounds of play. The opening round lasted from June 10^{th} until June 26^{th}. During this round the 32 participating teams were divided into eight groups. Each team then played one match against each of the other teams in its group. The top two finishers from each group qualified for the second round, known as the ‘Round of 16’. This round lasted from June 27^{th} through July 1^{st}. Beginning with this round the winner of each match advanced to the next round while the loser was eliminated. The remaining rounds of the tournament were: the Quarter Finals, held on July 3^{rd} and 4^{th}; the Semi Finals, held on July 7^{th} and 8^{th}; and the Final, held on July 12^{th}. A match to determine the third place finisher was held on July 11^{th} for the losing teams of the Semi Final round.

During the opening round each match was 90 minutes in length and was played in two 45 minute halves. During all subsequent rounds each match required a winner, so several tie breaking measures were used. If the match was tied after 90 minutes of regulation play, a 30 minute overtime period was played, with the first team to score declared the winner. If a winner had still not been determined, penalty kicks were used to decide which team would advance to the next round.

3 THE 1998 WORLD CUP WEB SITE

The Web site of the 1998 World Cup, `www.france98.com`, provided Internet-savvy football fans around the world with a wide range of information. Besides being able to access the current scores of the football matches in real time, fans could also access previous match results, player statistics, player biographies, team histories, information on the stadiums, facts about local attractions and festivities, as well as a wide range of photos and sound clips from the matches and interviews with players and coaches. Fans could also download free software, such as World Cup screensavers and wallpapers from the France ’98 Web site. All of the information on the site was available in English and French.

The France ’98 Web site went on-line May 6th, 1997. The site was established through the cooperative efforts of the Official Technology Suppliers to the World Cup: EDS, France Telecom, Hewlett-Packard, and Sybase. In anticipation of significant interest from the Internet community in this Web site, emphasis was put on deploying an available, reliable and low latency platform to power the Web site. During the tournament 30 servers were used, distributed across four locations: 4 servers in Paris, France; 10 servers in Herndon, Virginia; 10 servers in Plano, Texas; and 6 servers in Santa Clara, California. All of the Web pages were created and/or modified in France. New or updated pages were sent from France to the Plano site, which then distributed them to the other U.S. based locations. A Cisco Distributed Director was used to distributed client requests across the four locations. At each location various load balancers were used to distribute the incoming requests among the available servers.

4 COLLECTION AND REDUCTION OF DATA

The data set used in this workload characterization study is composed of the access logs collected from each of the servers used in the World Cup Web site. The access logs from each server were archived on a daily basis. For this study all of the access logs from May 1st, 1998 until July 23rd, 1998 were analyzed.

Each access log is in the Common Log Format [35]. For every request received by the Web server, the following information is stored:

```
remotehost rfc931 authuser [date] "request" status bytes
```

These fields are defined as follows:

- **remotehost**: the IP address of the client issuing the request
- **rfc931**: the remote loginame of the user
- **authuser**: the username as which the user has authenticated himself
- **[date]** : the date and time of the request

- **request**: the request line exactly as it came from the client
- **status**: the HTTP response status code returned to the client
- **bytes**: the content length of the document transferred

The **request** line from the client includes the method (e.g., GET, HEAD) to be applied to the requested resource, the name of the resource (e.g., /index.html), and the protocol version in use (e.g., HTTP/1.0).

An example of a (fabricated) access log entry is:

```
192.168.0.1 - - [10/Jun/1998:00:00:01 +0200] “GET /index.html HTTP/1.0” 200 1000
```

This entry tells us that on June 10th, 1998, at one second past midnight, local time in France, the client 192.168.0.1 asked this server for the file `/index.html`. The server was informed that the client supported HTTP/1.0. The server successfully responded to this request (this is indicated by the status code of 200) and transferred 1,000 bytes of content data to the client.

Table 1 summarizes the access logs that we acquired from the World Cup site. In total more

Table 1 Summary of Access Log Characteristics (Raw Data)

Duration	May 1st - July 23, 1998
Total Requests	1,352,804,107
Avg Requests/Minute	10,796
Total Bytes Transferred (GB)	4,991
Avg Bytes Transferred/Minute (MB)	40.8

than 1.35 billion requests were received by the Web site during the collection period, and

almost 5 TB of data sent to clients. The site averaged nearly 11,000 requests per minute and 41 MB of data were transferred to clients per minute on average.

Our first concern was with the size of the raw access logs - 125 GB in total, 14 GB when compressed. In order to make our workload analyses more efficient we chose to convert the logs to a more compact binary format. We reduced the storage requirements in two ways. One approach removed unnecessary data. For example, we deleted the **rfc931** and **authuser** fields as they were not used by the servers and thus provided no information that was of interest to us. The second tactic that we used to reduce the size of the data set was to represent the remaining fields in more efficient ways when possible. For example, we mapped all of the URLs to unique integers. We also mapped each distinct IP address to a unique integer identifier. Finally, we collated the access logs of all the servers by request time. The resulting binary log file was 25 GB in size, 9 GB when compressed. Furthermore, each request is now in a fixed size structure, which also helps to improve the efficiency of our analyses. Since all of the mappings we performed are reversible, we did not lose any (significant) information in the reduction process. There was some incorrect information in the raw access logs. For example, some of the status 304 replies included a non-zero response size. We left this incorrect information in the reduced log in case it is of interest for other researchers; we ignored it in our analyses.

Despite the vast amount of data that was collected by each of the servers, a lot of interesting information is still not available. For example, the access logs do not appear to contain information on the number of aborted connections that occurred. As a result, the number of bytes transferred reported in Table 1 overestimates the actual data traffic. The access logs have no information on either request or response header sizes which makes it impossible to know the total HTTP traffic for the site. Unfortunately the access logs have no precise information on when file modifications occurred. While the logs do have a timestamp that records when the request was received by the server, it has a one second resolution which is too coarse-grained to be of use for numerous analyses (e.g., inter-request times). These are just a few examples of useful information that could be added to a revised log file format.

5 WORKLOAD CHARACTERIZATION

This section presents the results of our workload characterization. Section 5.1 discusses various statistical characteristics of the data set, including the protocol version, method, response status code and file type distributions. Section 5.2 analyzes the usage of the World Cup site. Section 5.3 describes the file and transfer size distributions while Section 5.4 looks at the file referencing patterns. Section 5.5 investigates the usage of embedded files on the Web pages of the World Cup site. Section 5.6 presents an analysis of user sessions.

5.1 Statistical Characteristics

Our first analysis in this section looks at the version of the HyperText Transfer Protocol

Table 2 Breakdown of HTTP Version Supported by Client

HTTP Version	% of Requests	% of Content Data Transferred
0.9	0.00	0.00
1.0	78.66	79.83
1.1	21.32	20.09
x.x	0.02	0.08
Total	**100.00**	**100.00**

(HTTP) supported by the client issuing the request. Table 2 shows the results of this analysis. As expected, HTTP/1.0 is still the protocol used by most of the clients. However, the results indicate that a significant portion of the traffic, over 20%, came from clients that support HTTP/1.1. This indicates that browsers that support HTTP/1.1 are slowly replacing browsers that do not. These results do not indicate what percentage of the requests to the World Cup site, if any, actually used HTTP/1.1 functionality. We did not find any requests from clients that supported only HTTP/0.9. We did discover 270,561 requests (0.02%) that did not have a valid HTTP version entered in the access log. We chose to ignore this error as it will have no significant effect on our results.

Our next analysis looked at the resource method included in the each client request. Table 3 shows the distribution of requests by the method. 99.88% of all requests contained the GET method, which indicates that the stated URL is simply to be retrieved [19]. Included in this category are ‘conditional GETs’ (e.g., requests that include the If-Modified-Since header field) and ‘partial GETs’ (e.g., requests that include the Range header field) [19]. Unfortunately there is insufficient information in the log files to determine the exact number of conditional GETs that were issued by clients. This value is of interest as it would give us a better indication of how much impact client, proxy and network caching is having on the server workload. The next two most common methods seen were HEAD and POST. HEAD requests are issued when only the header of a file is desired and not the content. POST requests allow the client to send information to a specified URL on the server. A small number of other methods also appeared in the access logs, but not in sufficient numbers to affect the distributions given in Table 3. Table 3 indicates that some of the responses to HEAD requests appeared to have included content data which is a violation of the HTTP specification [19]. However, we ignore this as it has no noticeable impact on our study,

For the remainder of this paper we focus on analyzing the GET requests, as these account for almost all of the requests to the World Cup site. Since the primary purpose of this site was to provide information to people it is not surprising to see such a high percentage of requests include the GET method (i.e., the percentage of GET and POST requests is basically defined by the content on the Web site). We would like to point out that this will not be the case for all Web sites. In studies where methods other than GET are common, we would recommend analyzing all of the frequently used methods.

Table 3 Breakdown of Resource Methods

Method	**% of Requests**	**% of Content Data Transferred**
GET	99.88	99.62
HEAD	0.10	0.30
POST	0.02	0.08
Total	**100.00**	**100.00**

Table 4 shows the breakdown of server response codes. For a more complete description of possible response codes please refer to the HTTP/1.1 specification [19]. In this section we focus on the most commonly seen status codes in the access logs. Table 4 reveals that the majority of requests resulted in the Successful transfer of an object (response status 200). The Successful transfers account for almost all of the content data (97.86%) that was transferred from the Web site back to clients. The majority of the remaining content data traffic

Table 4 Breakdown of Server Response Codes

Response Code	**% of Requests**	**% of Content Data Transferred**
200 (Successful)	80.52	97.86
206 (Partial Content)	0.09	2.08
304 (Not Modified)	18.75	0.00
4xx (Client Error)	0.64	0.06
5xx (Server Error)	0.00	0.00
Other Codes	0.00	0.00
Total	**100.00**	**100.00**

was sent in status 206 (Partial Content) responses. The second most common status code was the Not Modified (304) response which accounted for almost 19% of all responses to client requests. This represents a substantial increase over the percentage of Not Modified responses seen in earlier server workloads [3]. The reason for this increase can be attributed to the improved caching architecture in the Web, including persistent caches in browsers, and more recently in proxies and networks (e.g., transparent caches). This type of response indicates that the client issued a conditional GET request to verify that its cached copy of the file is consistent with the version being served at the Web site. Since the Not Modified response is not the only possible response to a conditional GET request we still cannot determine the exact volume of conditional GET requests, although we can establish a lower bound. Relatively few requests resulted in error responses. Most of the errors that did occur were the result of incorrect URLs which resulted in a status 404 (File Not Found) response.

Table 5 shows the breakdown of response by the type of file requested by the client. The file

Table 5 Breakdown by File Type

File Type	% of Requests	% of Content Data Transferred
HTML	9.85	38.60
Images	88.16	35.02
Audio	0.02	0.10
Video	0.00	0.82
Compressed	0.08	20.33
Java	0.82	0.83
Dynamic	0.02	0.38
Other Types	1.05	3.92
Total	**100.00**	**100.00**

type was determined in several ways. For the majority of the responses the file extension was used to categorize the file by type. For example, files ending with '`.jpg`' or '`.gif`' were placed in the Images category, while files ending in '`.zip`' were placed in the Compressed category. We considered any URL that included '`cgi-bin`' in the string to be a dynamic response, as well as any file that had a '`.cgi`' or '`.pl`' extension. Furthermore, any URL that contained a parameter list (e.g., `/example.html?parameter_list`) was considered to be a dynamic file. For all of the remaining (unique) requests we issued HEAD requests to the Web site and used the `Content-type`: response header to classify the file.

Table 5 reveals that almost all client requests (98.01%) were for either HTML (9.85%) or Image (88.16%) files. A similar characteristic was observed in earlier Web server workloads [3]. Many of the remaining requests were for Java files. Few requests were made for multimedia types such as audio and video. HTML files had more impact than image files on the volume of content data transferred from the Web site (38.60% for HTML compared with 35.02% for Images). Most Image requests were for small inline graphics while the HTML requests were for substantially larger files. Furthermore many requests for Image files were conditional GETs that resulted in Not Modified responses. Thus many of the Image responses contained no data. The Compressed files, which accounted for only 0.08% of all requests were responsible for over 20% of the content data traffic. Most of the Compressed requests were for downloadable software, in particular World Cup screensavers for PCs. The huge discrepancy between the percentage of requests for Compressed files and the percentage of content data transferred for the corresponding transfers is an indication of the effects that large files can have on the workload of a Web server and of the network. The percentage of the content data transferred for Audio files does not indicate the actual impact on the network. Many of the Audio requests received at the Web site were for Real Audio files (i.e., streamed data). These requests were redirected to other servers. No information on these servers is available.

Table 6 shows the percentage of requests handled by as well as the content data transferred

Table 6 Breakdown by Location

Location	% of Requests	% of Content Data Transferred
Santa Clara, CA	16.60	16.89
Plano, TX	44.50	45.03
Herndon, VA	25.91	23.72
Paris, FR	12.99	14.36
Total	**100.00**	**100.00**

by each of the locations that participated in hosting the World Cup Web site. Client requests to the World Cup site were redirected to one of the four hosting locations by a Cisco Distributed Director. The goal of the Distributed Director is to transparently redirect client requests to the "closest" server. Closeness may be determined by client-to-server topological proximity or client-to-server latency [14]. However, both of these metrics are difficult to estimate. Table 6 indicates that most of the requests were handled by servers in North America, with the Plano location receiving the bulk of the work (44.50% of requests, 45.03% of content data).

The final analysis in this section examines the unique clients that accessed the World Cup Web site during the period of study. Determining an exact figure for the number of clients is virtually impossible. The presence of proxy caches and shared workstations hides some of the unique clients from our analysis. The use of DHCP to assign IP addresses to client machines inflates the number of unique clients seen in the log. Thus we can neither establish a lower nor an upper bound on the number of unique clients that issued requests.

The access logs contain 2,770,108 unique IP addresses. We analyzed each unique IP address to determine the number of requests they made to the World Cup Web site, the number of content bytes they received from the site, and the number of hosting locations that they communicated with during the trace. Table 7 presents the results of this analysis.

Table 7 Breakdown of Clients

	Location[a]	% of Unique Clients	% of Requests	% of Bytes Transferred
Single Location	SC only	10.49	2.75	2.67
	PL only	27.95	8.20	8.23
	HN only	14.23	3.98	3.60
	PA only	8.33	2.10	2.29
Subtotal		61.00	17.03	16.79
Two Locations	SC & PL	8.01	8.11	8.30
	SC & HN	0.79	0.47	0.43
	SC & PA	0.47	0.24	0.27
	PL & HN	9.35	9.97	9.36
	PL & PA	5.54	4.06	4.62
	HN & PA	2.42	1.43	1.44
Subtotal		26.58	24.28	24.42
Three Locations	SC, PL & HN	2.70	7.05	6.60
	SC, PL & PA	1.51	2.77	2.99
	SC, HN & PA	0.17	0.16	0.16
	PL, HN & PA	5.28	11.61	11.57
Subtotal		9.66	21.59	21.32
Four Locations	SC, PL, HN & PA	2.76	37.10	37.47
Total		**100.00**	**100.00**	**100.00**

a. Abbreviation definitions are given in Table 21, section 10 on page 87

Table 7 is divided into four parts, with each part providing information on the percentage of clients that communicated with servers at a particular location or number of locations. For example, the first section of Table 7 indicates the percentage of clients that communicated with servers at only a single location. 61% of all the unique clients communicated with servers at only one of the four hosting locations. These clients issued 17.03% of all requests to the World Cup Web site and received 16.79% of all content data transferred from the site.

From Table 7 we can see that most of the unique clients (87.58%) accessed servers at only one or two of the hosting locations. Assuming that the DistributedDirector is able to redirect clients to the closest hosting location this is not an unexpected behaviour. However, these clients are responsible for only 41.21% of all requests. 2.76% of all clients issued requests to all four hosting locations. Since these clients made over 37% of all requests it is likely that many of these clients are proxies. Although it seems counterintuitive that a client should communicate with servers from each location, particularly with the locations spread across two continents, we do not have sufficient information to properly evaluate the performance of the DistributedDirector.

5.2 Usage Analysis

Figure 1 shows the daily traffic volume handled by the World Cup Web site. From the begin-

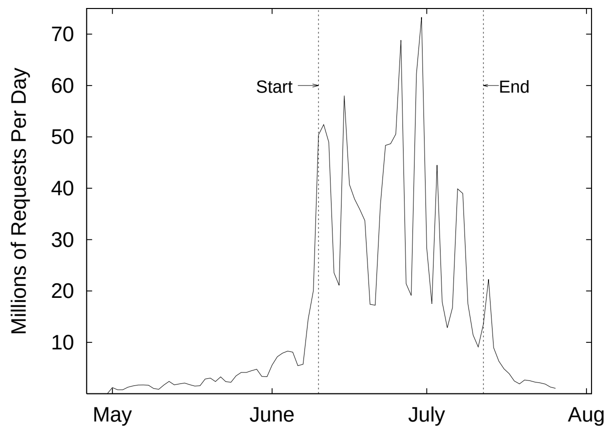


Figure 1 Daily Traffic Volume to the World Cup Web Site

ning of May until the start of the World Cup on June 10th the traffic volume is quite light although clearly building in anticipation of the start of the event. Beginning on June 10th the volume of traffic grows enormously. This marks the beginning of a prolonged “flash crowd”. That is, the site suddenly became very popular, remained popular for a lengthy period of time, and then just as quickly became relatively unpopular again. Although the daily traffic volume is quite bursty during the World Cup, the traffic volume remains higher than it was at any time prior to the start of the event. The busiest day for the site was June 30th when over 73 million requests were handled by the France ’98 site. After June 30th the daily traffic volumes begin to slowly diminish until the end of the World Cup, at which time the volume of traffic quickly subsides.

In order to better understand the causes of this burstiness we analyzed the traffic in more detail. Figure 2 shows the hourly traffic volume of the World Cup Web site.

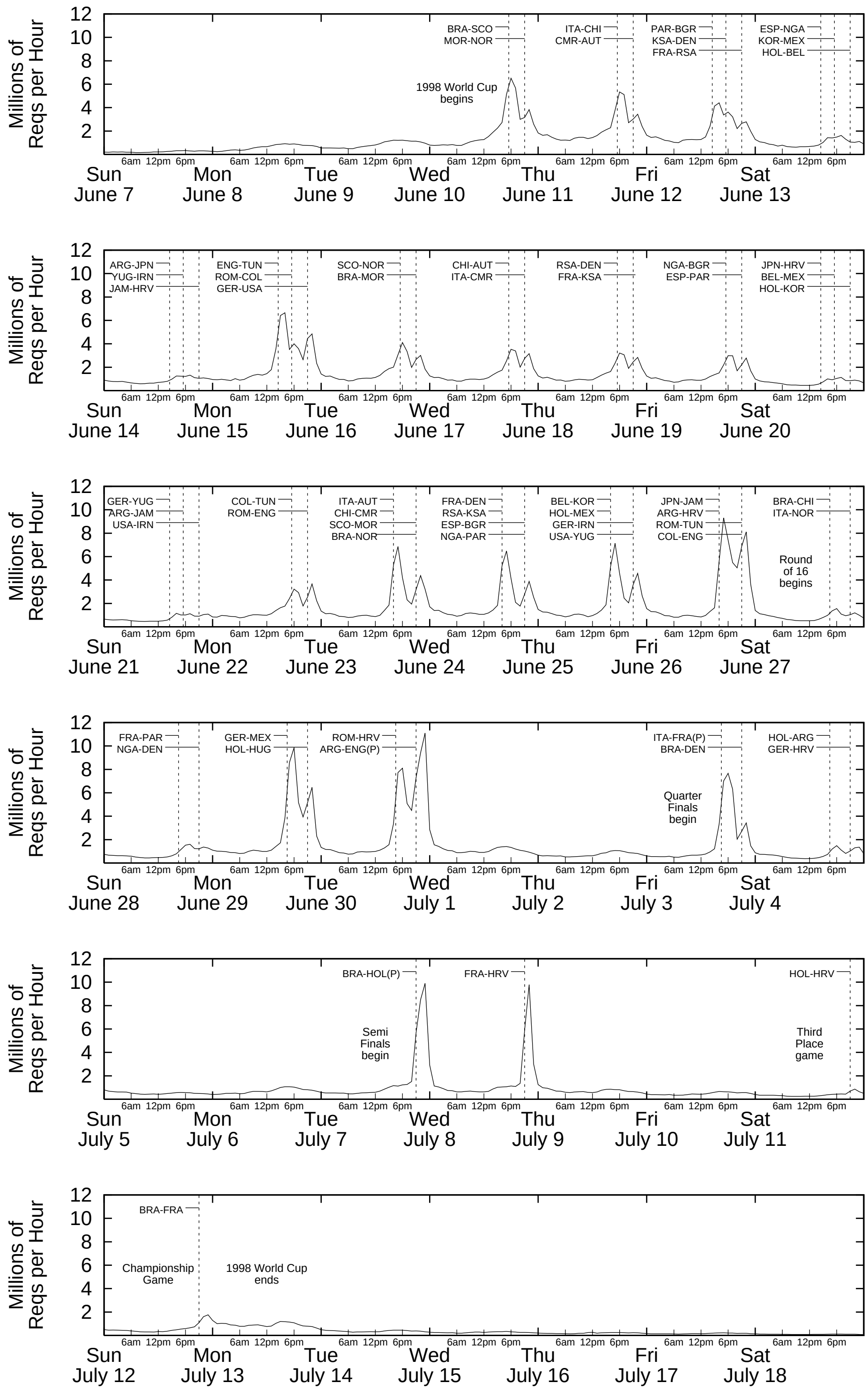


Figure 2 Hourly Traffic Volume to the World Cup Web Site

Figure 2 consists of six bar graphs, one for each week of the World Cup tournament. The solid black curve in each graph represents the hourly volume of requests (y-axis) for the given time (x-axis, normalized to local time in France). The scale of both the x and y-axes are kept constant across all bar graphs to facilitate comparisons in traffic volume over time and by day-of-week. The dashed vertical lines indicate the starting time of a World Cup football match. The teams involved in each match are also listed (the abbreviations are defined in Table 22 in Section 10). For example, at 5:30pm (in France) on Wednesday June 10th the first match of the 1998 World Cup was played between Brazil (BRA) and Scotland (SCO). Approximately six million requests per hour were received by the World Cup Web site at this time. The bar graphs also indicate the days on which each round of the tournament began (e.g., the Round of 16 began on Saturday June 27th), as well as those matches that required penalty kicks to decide a victor (e.g., on Tuesday July 7th the match between Brazil and the Netherlands was decided with penalty kicks; this is indicated by the (P) following the names of the teams).

Figure 2 reveals that there were many variables that affected the hourly traffic volume at the World Cup Web site. For example, the volume of traffic increased when matches were in progress and decreased once they had finished. These bursts represent flash-crowds on a smaller scale. The traffic volume was also affected by the teams involved in the matches (e.g., traditional football powers like Brazil and Germany are of interest to football fans everywhere and not just Brazilians and Germans), the number of matches in progress (e.g., from

June 23rd through June 26th matches were played in parallel), and the playoff implications of the match.

One interesting observation to be made from Figure 2 is that the volume of traffic to the World Cup Web site was quite low on weekends, even though a higher percentage of matches were played on Saturday and Sunday than on weekdays. The obvious reason for this reduction in traffic volume is that people preferred to watch the matches on television. When these fans were unable to watch the matches on television, such as when they were at work or school, or when certain matches were not televised in their area, they relied on the Web to provide them with progress reports on the matches that they were interested in. Timezone differences also contributed to the usage of the World Cup site. Since most of the matches were held in the late afternoon or evening local time this enabled European fans to watch most of the matches on television. Because of this we would expect to see different usage patterns for European-based clients accessing the World Cup site (e.g., fewer requests while matches are in progress). For fans in the Americas and (eastern) Asia this meant early morning or afternoon matches, which would often conflict with their daily routines.

As we mentioned earlier the busiest day for the World Cup Web site was June 30^{th}. Figure 2 provides us with an explanation of why this day was so popular. First of all, June 30^{th} was the last day of the Round of 16. Thus, the two victors on this day would advance to the Quarter Finals. Second, the match between Argentina and England went into overtime, and eventually required penalty kicks to determine the winner. During this match the request rate peaked at almost 12 million per hour.

5.3 Size Distributions

In this section we analyze the distribution of sizes for all unique files requested from the World Cup Web site. We also examine the distribution of all transfer sizes from the site.

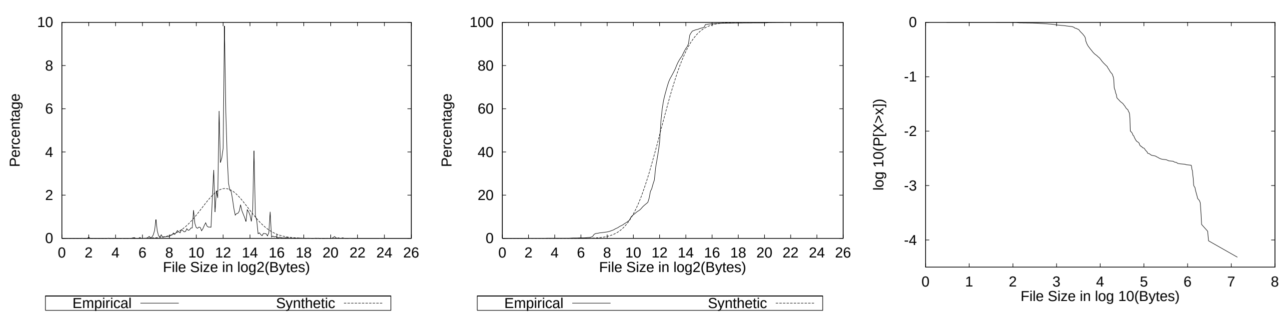
5.3.1 Size Distribution of Unique Files

Our first analysis looks at the sizes for each of the unique files that were requested and successfully transferred at least once in the access log. For the purpose of this study we utilize the initial non-zero size recorded for each unique file. Since some of the unique files change over time so too will the results of our analysis. However, we believe that the choice of which size to use for a file will only affect the parameters of the distribution and not the distribution itself. We have no information on the files that were available on the Web site but were not requested during the collection period.

Table 8 Unique File Size Information by File Type

	All Files	HTML	Image	Audio	Video	Java	Compressed	Dynamic
Number	20,728	11,411	7,025	344	12	14	67	1,783
Mean (bytes)	15,524	7,311	8,961	24,117	1,418,329	4,571	1,537,833	20,896
Median (bytes)	4,674	4,670	4,490	133	1,367,199	4,808	39,046	18,960
Maximum (MB)	61.2	0.14	1.32	1.33	1.86	0.006	61.2	2.9
Total Size (MB)	307	80	60	8	16	0.01	98	36

Table 8 presents some overall statistics on the unique files that were requested from the World Cup Web site. These statistics were calculated for the complete set of unique files (column entitled “All Files”) as well as for several different file types. Table 8 indicates that there were 20,728 unique files requested (and successfully transferred) from the World Cup site during the measurement period. The total combined size of these files was 307 MB. The mean size of these files was 15,524 bytes, the median size 4,674 bytes, and the maximum size 61.2 MB. Most of the unique files (18,436 of 20,728 or 89%) were in either HTML or Image format. An additional 9% of the unique files were considered to be dynamic (e.g., cgi-bin files). The unique HTML and Image files accounted for only 46% (140 of 307 MB) of the total size of the set of unique files. Much of the total size was due to a few large files, such as Video or Compressed. The Audio files were in general quite small. This occurred because many of the files in this category simply contained a URL that redirected the client to a Real Audio server. The few large Audio files were compressed sound clips (e.g., `.wav` files).



(a) (b) (c)

Figure 3 Size Distribution of Unique Files: (a) Frequency; (b) Cumulative Frequency; (c) Tail

Figure 3 shows the analysis of the size distribution for all unique files in the World Cup data set. Figure 3(a) presents the frequency histogram while Figure 3(b) provides the cumulative frequency histogram of the unique file sizes. We have applied a logarithmic transformation to the file sizes to enable us to identify patterns across the wide range of values [32]. For a $\log_2$ transformation, $\mathbf{bin_i}$ includes values in the range $2^i \le x < 2^{i+1}-1$. Similarly, for a $\log_{10}$ transformation, $\mathbf{bin_i}$ includes values in the range $10^i \le x < 10^{i+1}-1$. Figure 3(a) indicates that most unique files have sizes in the 256 byte to 64 KB range (2^8 - 2^{16} bytes). In other Web workload characterizations the file size distribution has been found to be *lognormal* [2][5]. That is, after applying a logarithmic transformation to the data, the data appears to be normally distributed. We compare the unique file size distribution (the *empirical data*) to a synthetic lognormal distribution with parameters μ=12.14 and σ=1.73. From Figure 3(a) we can see that the empirical data deviates quite substantially from the synthetic model. These differences are due to the distinct nature of the World Cup site. For example, in Figure 3(a) about 10% of all unique files were around 4 KB (2^{12} bytes) in size. 65% of these files are HTML objects that provided profiles on the individual players who participated in the World Cup tournament. All of the other large spikes in Figure 3(a) are also the result of groups of related objects having approximately the same size. On ‘typical’ Web sites we would not expect to see such large clusters of related objects that make up a substantial percentage of all files on the Web site.

Despite the number of spikes seen in Figure 3(a) the cumulative frequency histogram (shown in Figure 3(b)) indicates that the lognormal distribution still provides a reasonable estimate for the body of the unique file size distribution. While it is clearly not exact, the lognormal distribution may be sufficiently accurate for most modeling purposes.

While most of the unique files are less than 64KB in size a few are substantially larger. Our next analysis examined the tail of the unique file size distribution to determine if it is heavy-tailed. A distribution is considered heavy-tailed if $P[X > x] \sim x^{-\alpha}, x \to \infty, 0 < \alpha < 2$. This means that if the asymptotic shape of the distribution is hyperbolic it is heavy-tailed, regardless of the behaviour of the distributions for small values [12]. To determine if the unique file size distribution from the World Cup Web site is heavy-tailed we plotted the complementary distribution (CD) function on log-log axes and examined the results for linear behaviour on the upper tail. This method of analysis is described in [11]. The results of this analysis for the World Cup data are shown in Figure 3(c). The tail of the distribution does exhibit some linear behaviour which suggests that the distribution is indeed heavy-tailed. However, this linearity does not exist throughout the entire tail. Specifically, a spike exists in the 1-4 MB range. This spike is caused by the existence of 44 files whose sizes are in the 1-4 MB range. These files include 13 uncompressed, high resolution images, 4 audio clips, 15 screen savers (i.e., downloadable software) and 12 video clips.

To verify that the unique file size distribution is indeed heavy-tailed we utilized the scaling estimator tool `aest` created by Crovella and Taqqu [11]. This tool aggregates the data points in the distribution and then plots the complementary distribution of the aggregated data set. If the distribution is heavy-tailed then the tails of each successive aggregated data set will be approximately parallel with slope approximately -α [11].

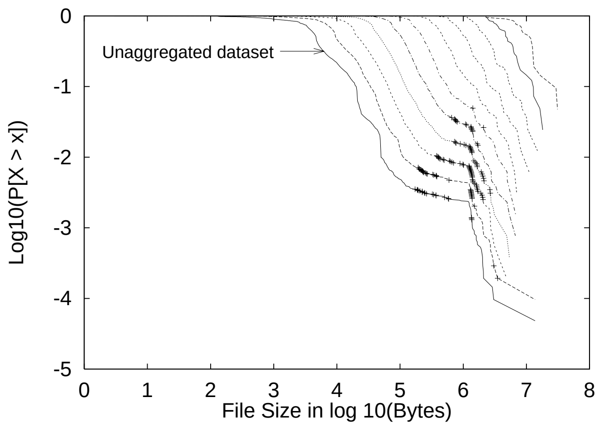


Figure 4 Size Distribution of Unique Files: Complementary Distribution Plots of Aggregated Data

Figure 4 presents the results of this test on the unique file size distribution for the World Cup Web site. The leftmost curve on the graph is the complementary distribution for the original, unaggregated data set. Each subsequent curve is the CD function for the data set that has been aggregated a factor of two more than the previous curve. The aggregation factors shown in Figure 4 are 1 (raw data), 2, 4, 8, 16, 32, 64, 128, 256, 512 and 1024.

The results in Figure 4 indicate that the unique file size distribution is heavy-tailed, as the tails of the successive CD plots are roughly parallel to one another. As the aggregation factor increases, the tails become linear throughout the tail of the distribution as the impact of the set of files in the 1-4 MB range diminishes. The estimate for the α parameter for the tail of this distribution is 1.37.

In summary, we believe that the unique file size distribution could be reasonably approximated using a hybrid model that combines a lognormal distribution for the body and a power law distribution for the tail. This is the approach taken with SURGE, a Web workload generator developed by Barford and Crovella [6]. A more precise model would need to account for the clusters of related files found in the workload which affects both the body and tail of the distribution.

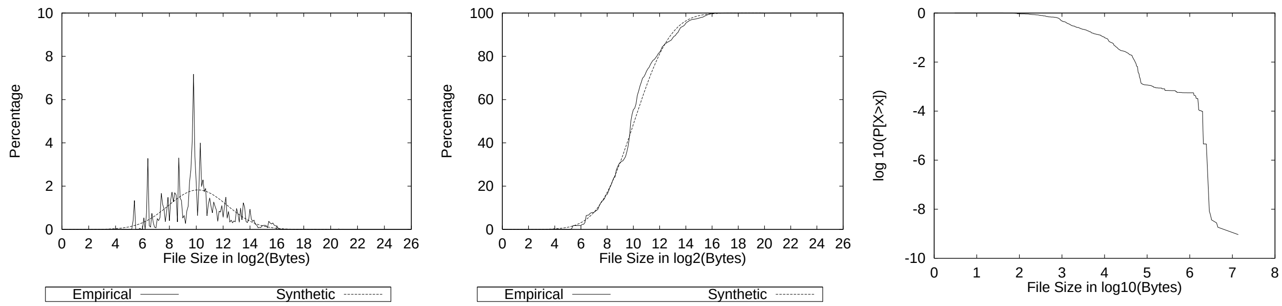
5.3.2 Size Distribution of Successful Transfers

Our next analysis focuses on the sizes of all successful transfers (i.e., status 200 responses) from the World Cup Web site.

Table 9 Successful Transfer Size Information by File Type

	All Transfers	HTML	Image	Audio	Video	Java	Compressed	Dynamic
Number	1,087,916,098	107,312,796	946,428,396	281,149	28,600	10,139,230	969,058	282,615
Mean (bytes)	4,802	18,693	1,965	19,370	1,464,641	4,367	1,018,305	72,323
Median (bytes)	965	12,624	914	131	1,367,199	4,406	1,272,120	6,122
Maximum (MB)	61.2	0.23	1.32	1.33	1.86	0.01	61.2	4.32
Bytes Transferred (GB)	4,856	1,868	1,732	5	39	41	918	19

Table 9 presents the overall statistics on the successful transfers, for all transfers and by file type. By comparing Table 8 and Table 9 we can see numerous differences between the unique file and successful transfer size distributions. For example, the median successful transfer size is 965 bytes, which is significantly smaller than median of 4,674 bytes for the unique file size distribution. This difference indicates that the smaller files available at the site were requested significantly more often than the larger files. For HTML files the median transfer size is larger than the median unique size. This occurred in part because the more popular HTML pages were quite large, and because some of the HTML pages increased in size during the data collection period. Evidence of these increases can be seen by comparing the maximum (initial) size for all HTML files and the maximum transfer size seen for HTML files.



(a) (b) (c)

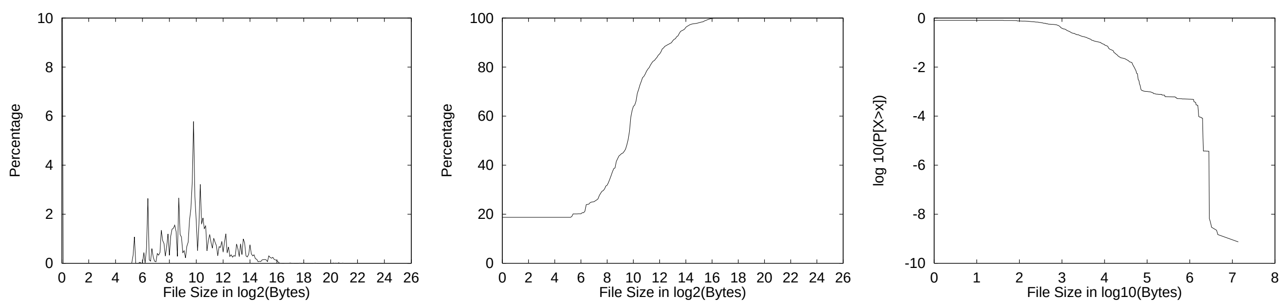
Figure 5 Size Distribution of Successful Transfers: (a) Body; (c) Tail

Figure 5(a) shows the frequency histogram for the successful transfer size distribution. As was the case for the unique file size distribution, the frequency histogram contains a number of large spikes. This characteristic is due to several of the more popular objects having very similar sizes. The cumulative frequency histogram, shown in Figure 5(b), reveals that the synthetically-generated lognormal distribution with parameters μ=10.13 and σ=2.19 is a much better model for the successful transfer size distribution than it is for the unique file size distribution.

Figure 5(c) presents the analysis of the tail of the successful transfer size distribution. In Figure 5(c) we can see that the tail is affected by transfers in the 1-4 MB range just as the tail of the unique file size distribution was affected by files in this range (Figure 3(c)). For the Successful Transfer case the tail of the distribution is not affected by the number of files in this size range but rather by the popularity of several of these large files [10]. For example, the five most popular files in this set of large files were World Cup screen savers that people could download and use on their PCs. These five files were transferred over 600,000 times during the period of data collection.

5.3.3 Size Distribution of All Transfers

In this section we examine the size distribution of all transfers from the World Cup Web site.



(a) (b) (c)

Figure 6 Size Distribution of All Transfers: (a) Body; (b) Tail

Figure 6 presents the analysis of the body and tail of the size distribution for all transfers. Figure 6(a) and Figure 6(b) show the frequency and cumulative histograms respectively for the overall transfer size distribution. The spike at 0 in this graph corresponds to the high volume of Not Modified responses seen in the workload (we placed all zero-sized transfers in the 2^0 bin since $\log_2 0$ is undefined). The presence of this large quantify of zero-sized transfers reduces the median transfer size to 828 bytes from 965 bytes for the Successful transfers. This spike is the main difference between the Overall Transfer size distribution and the Successful transfer size distribution.

5.3.4 Impact of Size Distributions

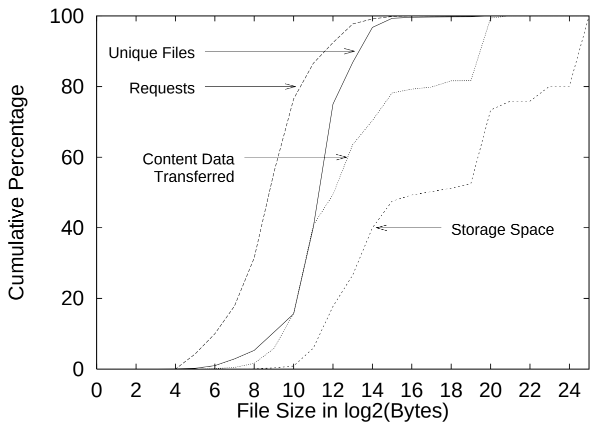


Figure 7 Impact of File Size Distributions on Servers and Network

Figure 7 indicates the effect of the size distributions on the storage requirements at the World Cup Web site as well as on network traffic. For example, files up to, but not including 1 KB in size (up to and including 2^9 bytes) account for 10.4% of all files stored at the World Cup site but utilize only 0.4% of the storage space at the site. 55.7% of all client requests were for files less than 1 KB in size. Responses to these requests generated only 5.8% of the total content data transferred from the Web site. Meanwhile, files 64 KB and larger (2^{16} bytes) made up only 0.4% of the unique files but consumed 50.7% of the required storage space. Although these files received only 0.1% of all client requests they accounted for 21% of the content data transferred. These numbers suggest that the impact that these few large files can have on the system is substantial. Thus it is important to accurately model the upper tails of both the unique file and transfer size distributions in order to make better assessments of the impact of a workload on a Web server [6].

5.4 File Referencing Behaviour

In this section we analyze the World Cup workload for the presence of two important file referencing characteristics: temporal locality and concentration of references.

5.4.1 Temporal Locality

Temporal locality means that a file that was recently referenced will likely be referenced again in the near future [1][3]. To measure the temporal locality we utilize the standard LRU (Least Recently Used) stack-depth analysis. This analysis works in the following manner. When a file is initially referenced it is added to the top of the LRU stack (position 1). All files that are currently in the stack are pushed down by one position. When a file is referenced again its current depth (i.e., position) in the stack is recorded and then the file is moved back to the top of the stack. The other files in the stack are pushed down as necessary. Once the entire log has been analyzed the record of the depths at which re-references occurred is examined. Logs which exhibit a high degree of temporal locality will have a small average (or median) stack depth. Conversely, logs with a low degree of temporal locality will have a large mean (or median) stack depth.

Table 10 Temporal Locality Analysis

	All Locations	Santa Clara	Plano	Herndon	Paris
mean stack depth	290	272	261	261	414
standard deviation	721	637	621	639	1073
median stack depth	106	107	101	97	137
90th percentile	615	589	564	559	816
normalized mean stack depth	0.015	0.014	0.014	0.014	0.022
normalized median stack depth	0.0051	0.0052	0.0048	0.0047	0.0066

Table 10 shows the results of the stack depth analysis for the World Cup workload. We performed the analysis for the site as a whole (i.e., considering all requests) and for each location independently. Similar mean stack depths in Table 10 indicate that the degree of temporal locality is quite consistent across the three North American locations. The degree of temporal locality is noticeably weaker at the Paris site. This difference is due to the Paris site having to serve both French and English pages on a regular basis. The US based locations typically received requests only (although not exclusively) for pages in English.

In Table 10 we also provide the median stack depth. Across all sites the median stack depth is significantly smaller than the mean, indicating that the degree of temporal locality in the workload is even stronger than is suggested by the mean depth. This observation suggests that the stack depth distribution has an extremely long tail. Further evidence of this is provided by examining the 90th percentile. Across all of the server locations 90% of the references were at a depth of 816 or less, which is only 4% of the maximum depth of 20,728 (the number of unique files in the trace).

Table 10 also includes the normalized mean and median stack depths. We calculated these values by dividing the mean or median stack depth by the number of unique files in the trace. By normalizing the stack depth we can compare the degree of temporal locality across different access logs [5]. For example, Barford *et. al.* reported a normalized mean stack depth of 0.2340 and a normalized median stack depth of 0.0399 for a recent proxy trace [5]. The normalized mean and normalized median stack depths reported in Table 10 are significantly smaller than the values reported by Barford indicating (as expected) that the temporal locality is much stronger in the Web site accesses.

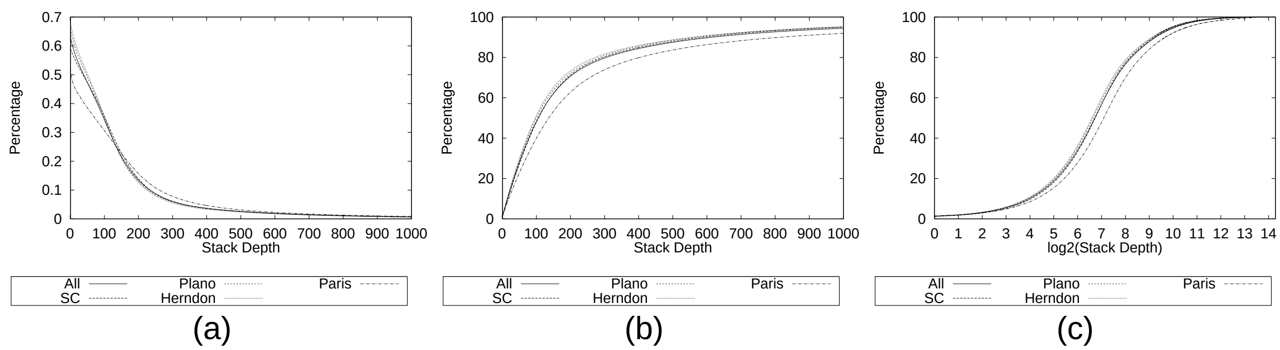


Figure 8 Stack Depth Distribution: (a) Frequency; (b) Cumulative Frequency; (c) Log-Transformed Cumulative Freq.

Figure 8(a) and Figure 8(b) provide the frequency and cumulative frequency histograms for the stack depth distributions. The results for the overall workload as well as for each location are presented. These two figures clearly show that most of the references occurred near the top of the stack, indicating a strong degree of temporal locality. In fact, 94% of all references occurred at a stack depth of less than 1000. These figures also show, as we indicated earlier, that the stack depth distributions are quite consistent across the three North American locations but noticeably different for the Paris location. In order to view the entire stack depth distribution we applied a logarithmic transformation to the data. The results are shown in Figure 8(c). The remaining 6% of references occurred in the bottom 95% of the stack (positions 2^{10} and greater).

5.4.2 Concentration of References

The second file referencing characteristic that we focus on is concentration of references. Many studies, including [3] and [13], have found that a non-uniform referencing pattern exists for files on the World-Wide Web. This means that a small number of files on a Web site are extremely popular and are responsible for most of the requests arriving at the site. Most of the unique files on a Web site are unpopular and are seldomly requested.

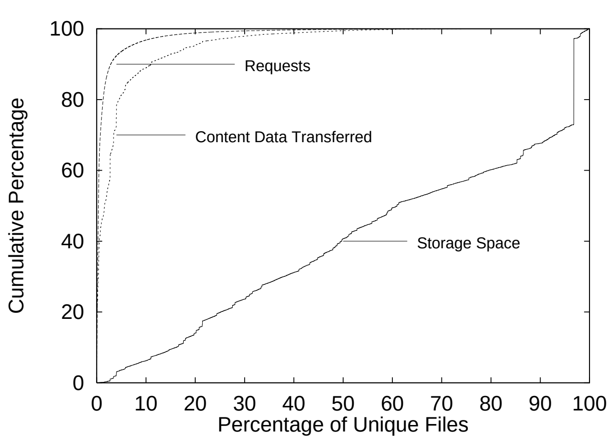


Figure 9 Concentration of References (cumulative distribution)

Figure 9 shows the distribution of all client requests across the set of unique files available at the World Cup Web site. In this figure all of the unique files (x axis) have been sorted in decreasing order by the number of references that each received. The volume of content data that each of these files generated in network traffic along with each file's storage requirements at the site were also computed. Figure 9 clearly shows that there is a concentration of references among a small subset of the unique files. For example, the top 10% (i.e., the most popular) unique files received 97% of all requests, generated 89% of the network traffic (i.e., content data) while occupying less than 7% of the storage space on the Web site. The top 1% of files received 75% of all requests, generated 46% of the network traffic and consumed a mere 0.12% of the required storage space.

While a number of the files on the World Cup site were extremely popular, many were relatively unpopular. In fact, 9.2% of the unique files were requested only a single time. We refer to these files as "one-timers" [3]. The combined size of these one-timers was 98 MB, or 31.8% of the combined size of all unique files. This characteristic is of interest because of its obvious effect on caching; even over a long period of time and with an exceptionally heavy workload, some files on a Web site will not be referenced more than once. Thus, there is no benefit in caching these files.

Several studies, including [3], [5], [7] and [13], have found that a Zipf(-like) distribution can be used to characterize the popularity of files on the Web. A distribution is considered to be Zipf-like if the relative probability of a request for the $\mathbf{i^{th}}$ most popular object is inversely proportional to $\mathbf{1/i^{\beta}}$ [7]. In Web proxy workloads, estimates of β typically range from 0.5 to 1

[5][7][27][33]. The more concentrated the references are to a set of popular objects, the higher the estimate of β. Since the concentration of references in Web server workloads is generally much stronger than what is found in Web proxy workloads, the estimates of β are also higher.

To test if a distribution is Zipf-like a log-transformed plot of the number of requests for each file as a function of the file's rank is created. The most frequently requested file is assigned a rank of 1 while the least frequently requested file is assigned rank *N* (in this case 20,728). If the distribution is Zipf-like the graph should appear linear with slope near -β [5].

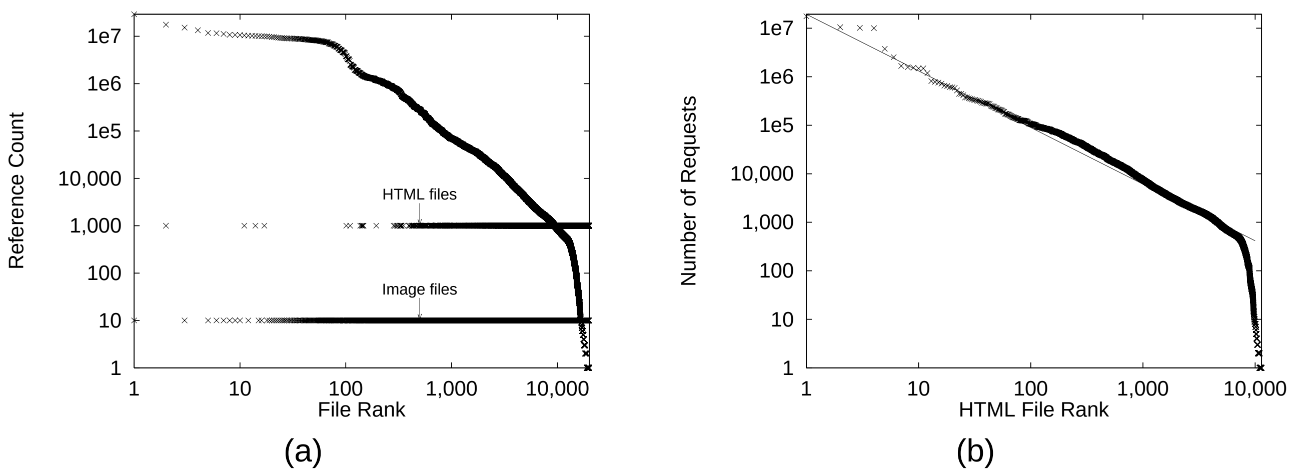


Figure 10 Concentration of References (reference count vs. rank): (a) All files; (b) HTML files only

Figure 10(a) shows the relative popularity of the unique files in the World Cup workload. This graph exhibits three distinct linear regions; thus it does not appear to be Zipf-like when considering all unique files. The three distinct regions are (I) files 1-100, (II) files 100-13,000, and (III) files 13,000 to 20,728. Figure 10(a) also includes information that indicates which files are HTML and which are images. The two horizontal lines at the bottom of Figure 10(a) indicate if the file was an HTML or an Image file. For example, these lines can be used to determine that the most popular file (rank 1) was an image while the second most popular file (rank 2) was an HTML file. These lines also reveal that most of the top 500 files were images and only a few were HTML. Unfortunately these lines are too dense to distinguish the types of the less popular files.

We will now examine each of the regions mentioned above in more detail. In region I the slope of the graph is nearly horizontal (slope is approximately -0.25), indicating that all of the files have nearly equivalent reference counts. These 100 files received 61% of all requests to the World Cup site and caused 37% of all content data traffic. Of these 100 files six were HTML and 93 were images. We speculate that the reason there are so many files with nearly equivalent reference counts is due to the number of embedded files in the popular HTML files (58 of the 93 Image files were embedded in the 6 HTML files in region I), and the use of the same images across many different pages on the World Cup site (the remaining Image files in region I were embedded in multiple pages; 13 of these Image files were utilized on 400 or more pages). Caching within the network (e.g., at clients and proxies) may have reduced the reference counts of some of the more popular files. The graph in region II of Figure 10(a) is linear with a slope estimated at -1.92. In region III the graph drops off almost vertically (slope estimated at -14.7). We are unsure of the cause of this. One hypothesis is that the extreme popularity of the World Cup site and the relatively small set of unique files changes the behaviour of the distribution.

Since the popularity distribution for all unique files did not appear to be Zipf-like we decided to perform the same analyses just on the HTML files. We chose to examine the HTML files as this would provide us with an estimate of the popularity of the pages on the World Cup Web site (due to the use of Frames on this site, a number of pages actually consist of multiple HTML files). The results of this analysis can be seen in Figure 10(b). Two distinctive regions can be seen in this graph. In region I (files 1 - 6,000) the popularity of HTML files appear to follow a Zipf-like distribution reasonably well. We estimate the slope of this portion of the graph at approximately -1.16. In region II (files 6,000 - 11,411) the graph drops off almost vertically, with a slope estimated at -20.6. We are unsure of the cause of this change in the graph. One possibility is that many of the HTML files in region II were available (i.e., linked to other pages) for only short periods of time, or perhaps not at all. Once a file is no longer linked to other files it can only be accessed by directly requesting it (i.e., typing in the URL of the file). This would significantly reduce the number of accesses to the file.

5.5 Embedded Files

In an updated version of the SURGE workload generator [6], Barford and Crovella define three classes of files [4]:

- **base files**: HTML files which contain embedded files
- **embedded files**: files which are referenced by base files (e.g., images, java)
- **single files**: files which are neither base nor embedded (e.g., compressed)

For simplicity we assume all HTML files are base files, all images and java files are embedded files, and all other types are single files.

In this section we focus on the embedded files. In particular we want to determine the distribution of total embedded files per base file, as well as the distribution of unique embedded files per base file. We also examine the use of individual embedded files across multiple base files.

The total number of embedded files in a base page represents the upper limit on the number of additional HTTP requests that will be generated whenever the base file is requested. Due to caching by the browser additional HTTP requests should only be needed for the unique embedded files referred to by the base file. Because some files may be embedded in more than one base file the actual number of additional HTTP requests that are automatically generated when a particular base file is requested should be less than the number of unique embedded files contained in that base file. However, this distribution is affected by the cache size and consistency policy at the client and is therefore more difficult to quantify.

We did not utilize information from the log files to determine the number of embedded files per base file. Instead we analyzed a copy of the World Cup site. We set up a local Web server to host the files from the site. We then utilized the remote control feature of the Netscape Navigator browser [31] to request each base file. We used the browser to interpret the Javascript in the base files and to request the appropriate embedded files (we took several steps to ensure that the browser would issue requests for all of the embedded files rather than files from its cache). This step was required as the number of embedded objects per base file depended on the capabilities of the client's browser. Simply counting the number of embedded files in each HTML file would overestimate the number of embedded files utilized for a particular browser (e.g., simply scanning the HTML files resulted in a maximum of 76 embedded files compared to a maximum of 61 using Netscape to generate the requests). The results we report are for Mozilla 4.0 (i.e., Netscape 4.0) compliant browsers. We believe that fewer embedded files were utilized for older browsers although we have not analyzed this thoroughly. Finally we analyzed the access log of our Web server to determine the embedded files for each base file.

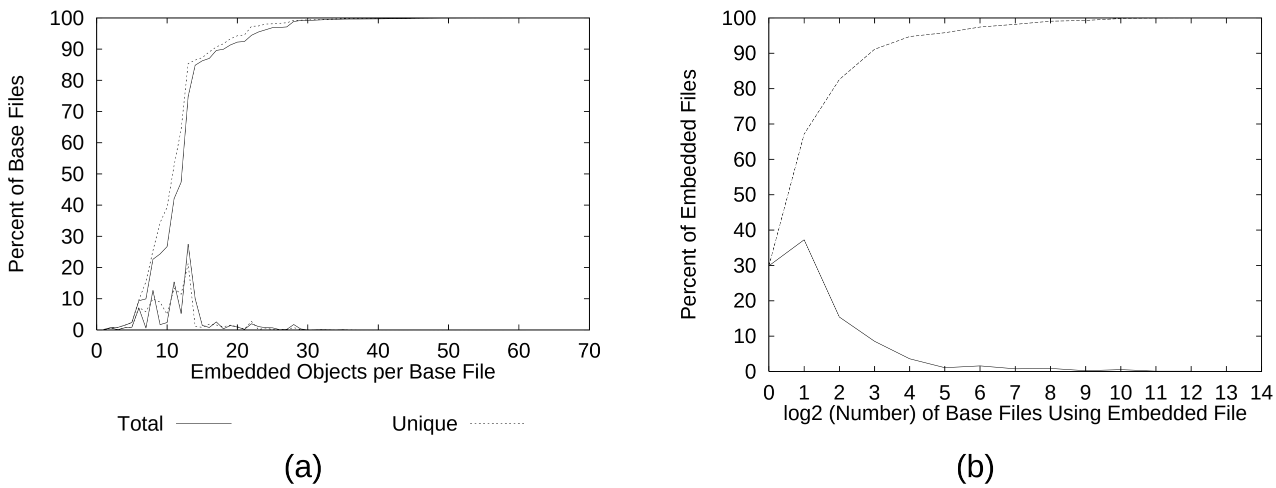


Figure 11 Analysis of Embedded Files: (a)Embedded Files per Base File; (b) Sharing of Embedded Files

Figure 11(a) shows the distributions for the total embedded files per base file as well as for the unique embedded files per base file for the World Cup Web site. 90% of the base files had a total of 19 or fewer embedded files. The median value was 13 total embedded files per base file. The maximum number of embedded files on a single base file was 61. Since some embedded files are used more than once in a single base file we also analyzed the distinct embedded files per base file. When only the unique embedded files are considered the numbers are slightly smaller; 90% of the base files included 17 or fewer embedded files, while the median value was 11. The maximum number of unique embedded files was 58.

Figure 11(b) shows the number of base files that an individual embedded file is likely to be included in. Most of the embedded files are included in only a few base files. For example, 90% of all embedded files are used in 15 or fewer base files. Included in this group of embedded files are the pictures of individual players. Many of these images appear in only a single base file, namely the biography page for the particular player. While most embedded files appear in only a few base files, a small number of embedded files are widely used. The most popular embedded file, a small icon, appears on 7,969 of the 11,411 HTML files.

5.6 User Session Analyses

In this section we investigate various characteristics of user sessions. For the purpose of these analyses we define a user session as all requests from a single client to the World Cup Web site, with the time between requests from that IP address less than some threshold value. That is, if request r_{i+1} from client C arrives at the Web site x seconds after request r_i from client C, and $x \leq t$ (t is the timeout value in seconds) then requests r_i and r_{i+1} are both considered to be part of session s_n for client C. If $x > t$ then request r_i is deemed to be the final request of session s_n for client C, while request r_{i+1} is the initial request of session s_{n+1} for client C.

We consider each unique IP address in the access log to be a distinct client or user. Clearly this is not true in all cases. For example, some of the IP addresses in the access log belong to proxies which issue requests on behalf of multiple users. The presence of proxies in the data set can reduce the estimates of the number of unique users of the site and alter the characteristics of user sessions. It is also possible that some unique users utilize multiple IP addresses (e.g., using different computers to access the Web, or receiving a different IP address via DHCP when connecting to the Internet). The main effect of this is an inflation in the estimated number of unique users. Non-human users such as Web crawlers may also be present in the access logs. The behaviour of these type of clients is quite different from human users and will result in different session characteristics. However, based on the results from Section 5.2 we believe that most of the traffic to this site was generated by human users. Thus we make no attempt to identify or remove requests that may have been generated by agents such as Web crawlers. Also, we have no information on whether persistent connections were enabled on the World Cup servers.

Although estimates of the number of unique users and the cumulative number of users that visited the World Cup Web site are of interest, our focus in this section is on user session characteristics and the possible implications on HTTP behaviour. In particular we concentrate on evaluating (at a high level) the effectiveness of persistent connections in reducing the number of TCP connections required for client-server communication on the Web. By reducing the number of TCP connections persistent connections reduce user latency by eliminating unnecessary round trips for the establishment of TCP connections. Persistent connections are also able to avoid latency associated with TCP slow start under certain conditions [4][30]. One disadvantage of persistent connections is the need for the server to maintain a much larger number of open TCP connections. We estimate this effect by monitoring the number of *active sessions* at the World Cup Web site. We consider a session to be active if the client has issued at least one request within the last ***t*** seconds (i.e., the session has not timed-out at the server). Since we are evaluating persistent connections at a high level we do not investigate the effects of pipelining requests within a persistent connection. Our goal is to get an initial indication of the effectiveness reusing TCP connections for this workload. We realize that we will be underestimating the number of connections that a server would have to keep state on, since a server must maintain state for a period of time after the connection has been closed. This more precise analysis is left for future work.

In the remainder of this section we examine the effects of various timeout values on the total number of user sessions in the World Cup workload, the maximum number of active sessions, the length of sessions, the number of requests per session, and the time between sessions.

5.6.1 Total Sessions

Our first analysis looks at the total number of sessions and the maximum number of active sessions that occur for a wide range of timeout values. There are two extreme cases to be aware of. If no reuse of TCP connections happens (as is the case with HTTP/1.0, ignoring KeepAlive connections), 1,351,193,319 sessions would occur, one for each GET request. In this case relatively few sessions would be active simultaneously (during the busiest period of the workload requests arrived at a rate of 3,600 per second). The other extreme happens when each client receives a persistent connection that is held open indefinitely. In this situation 2,770,108 sessions would occur, one for each unique client in the access log. This represents only 0.2% of the sessions that occur in the other extreme, although the site is now required to maintain state on three orders of magnitude more active sessions.

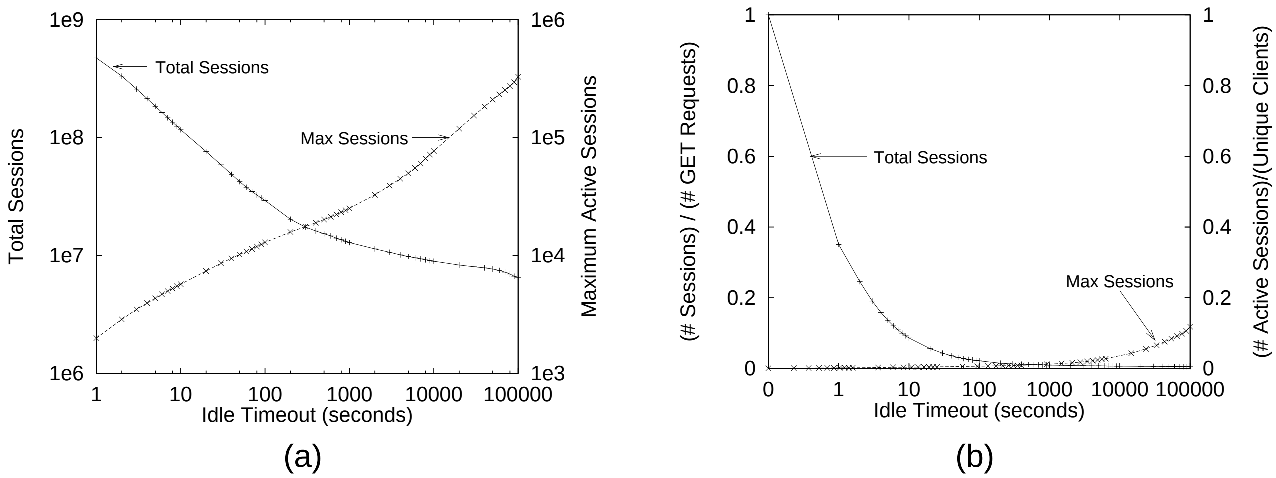


Figure 12 Effect of Timeout Values on Total Number of Sessions

Figure 12 shows the effects that different timeout values have on the total number of sessions and on the maximum number of active sessions seen in the World Cup workload. The results are quite similar to those reported by Mogul [30]. Figure 12(a) shows the actual number of sessions that occur for a given timeout value. As the timeout values increase the total number of sessions drops rapidly. For example, with a timeout value of 100 seconds, the number of observed sessions is 29,249,442 compared to 1.35 billion sessions when no reuse occurs. Once timeout values larger than 100 seconds are used there is little further reduction in the total number of sessions, even with substantial increases in the timeout value. However, the maximum number of active sessions grows quite rapidly with increases in the timeout threshold. Figure 12(b) shows the results of this analysis as a fraction of the extreme case (i.e., one session per request). For example, with a 100 second timeout only 29 million sessions, or 2.2% of the maximum 1.35 billion sessions occur. The maximum active sessions for this timeout value is 12,890, or 0.47% of the maximum of 2.8 million.

5.6.2 Active Sessions

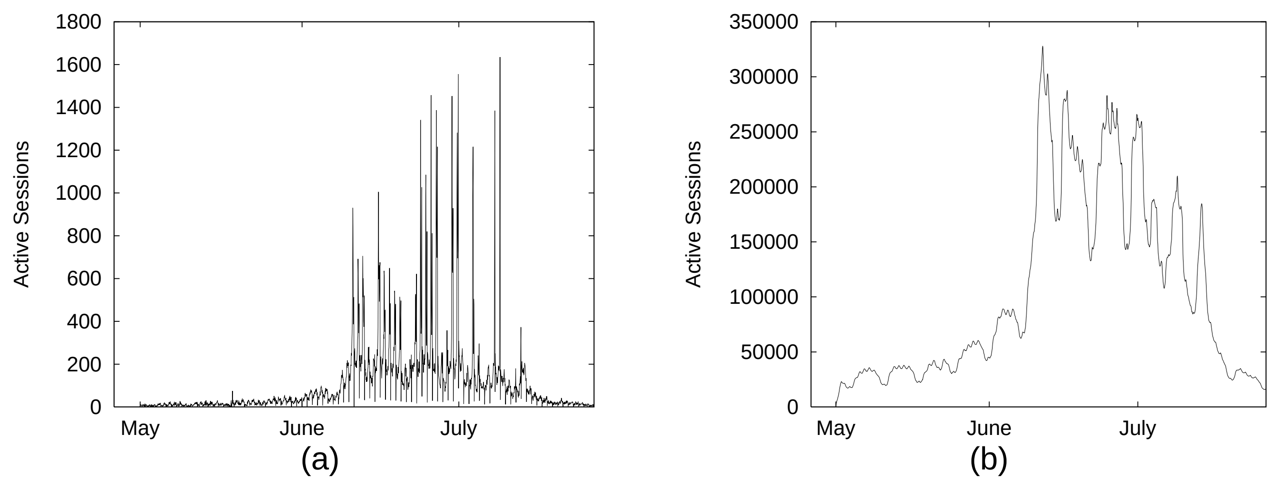


Figure 13 Active Sessions over Time: (a) 1 second Timeout; (b) 100,000 second Timeout

In the previous subsection we discussed the maximum number of active sessions that occurred for various timeout values. In this section we analyze the number of active sessions over time. Figure 13(a) shows the number of active sessions reported at the beginning of each one hour interval over the entire World Cup workload. In this graph a one second session timeout is used. As expected the number of active sessions is very bursty, The spikes in Figure 13(a) increase in size as the World Cup tournament progressed. The largest spikes correspond to the two semi final matches. The results change somewhat as larger timeout values are used. Figure 13(b) shows the number of active sessions each hour when a 100,000 second timeout is used (slightly more than one day). This graph is still bursty although much less so than with smaller timeout values as many of the short sessions from clients who visited the site multiple times have been merged into a few longer sessions. Perhaps the biggest difference though between Figure 13(a) and Figure 13(b) is the trend in the size of the spikes, which are now decreasing in size over time. This suggests that the number of people visiting the site decreased over time although those who remained visited more frequently and for shorter durations.

5.6.3 Session Length

Our next analysis looks at the effect of the timeout value on the length of sessions. We calculate the session length as the time between the arrival of the first request and the arrival of the last request in the session. The session length does not include the timeout value. Excluding the timeout value allows us to see how long the clients are using the sessions. To determine how long the server would need to maintain the session simply shift each curve by the timeout value. Since the access logs do not include any information on the time needed for the server to complete the response our results will underestimate the session lengths, particularly for the shorter timeout values.

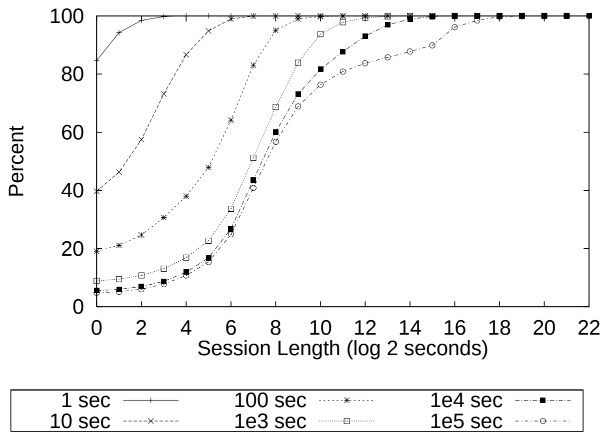


Figure 14 Analysis of Session Lengths

The results of this analysis are presented in Figure 14. As expected the session lengths increase with longer timeout thresholds. For example, with a one second timeout 85% of the sessions lasted only a single second. When the timeout value is increased to 100 seconds 81% of the sessions lasted longer than one second, with 52% lasting longer than 64 seconds. As the timeout values increase beyond 1,000 seconds the bodies of the session length distributions change very little. However, the tails of these distributions get longer and longer. We assume that this is caused by the presence of proxies in the access log. The 20% tail of the 100,000 second timeout curve is quite different from all of the other curves. The cause of this is the group of clients, presumably diehard football fans, that retrieved information from the site on a daily basis. Once the timeout value exceeded the time between the daily sessions of these clients a few extremely long sessions were created. The longest session length calculated was 49 days.

5.6.4 Sessions Per Client

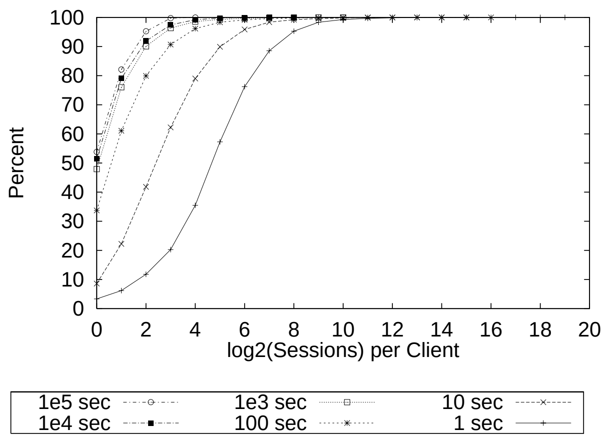


Figure 15 Analysis of the Number of Sessions Per Client

Figure 15 shows the distribution of the number of sessions that each client had for the range of timeout values examined. From Figure 15 we can see that as the timeout value increases, the number of sessions per client drops substantially. For example, with a one second timeout 65% of clients had more than 16 sessions (2^4) during the course of the World Cup. As the session timeout increases to 100 seconds, only 40% of clients had more than 16 sessions. Increasing the session timeout value beyond 1,000 seconds decreases the number of sessions only slightly.

5.6.5 Requests and Bytes Transferred per Session

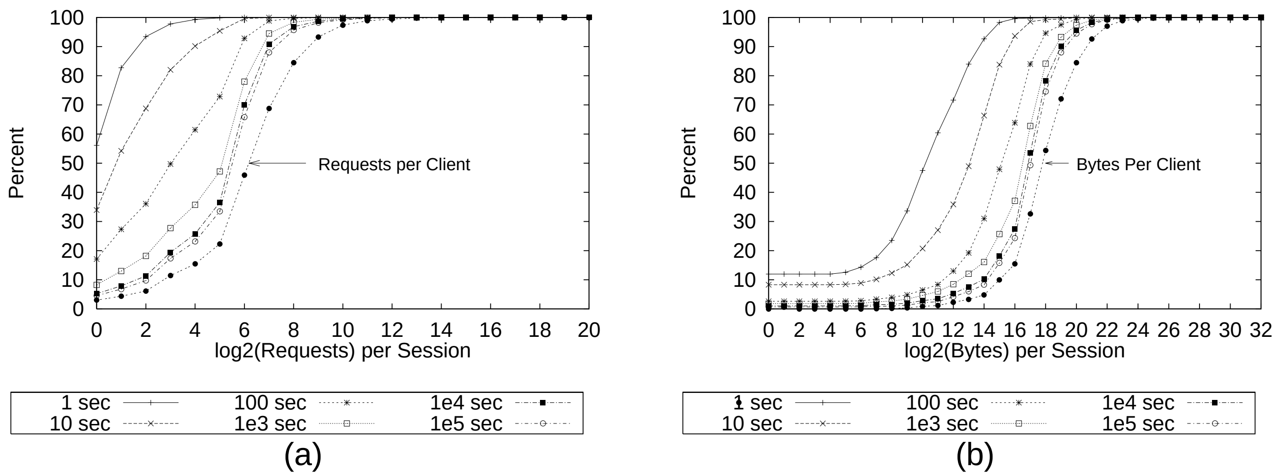


Figure 16 Analysis of per Session Activity: (a) Requests; (b) Bytes Transferred

In this subsection we analyze the number of requests issued by each client as well as the number of (content data) bytes transferred to each client during a session. Obviously these numbers will tend to increase as the timeout value (and session length) grows. The results of this analysis are shown in Figure 16. The right most curve in each graph indicates the distribution of requests or bytes transferred when exactly one session is used for each unique client. Thus this curve reveals the highest utilization of persistent connections that could have occurred for this workload (i.e., this is the best case scenario; once a session is established it never times out) The other curves on the graphs indicate the distributions for the various timeout values that we examined. For timeout values of 1,000 seconds or more the distributions are becoming very close to the best utilization that we could expect to see.

Figure 16(a) indicates the number of requests per session for the different timeout values. One intriguing observation from this graph is the percentage of sessions during which the client issues only a single request. Even though the percentage of sessions that exhibit this behaviour decreases rapidly as the timeout value increases, 17% of sessions (when using a 100 second timeout) sent only a single request to the World Cup site. To determine the cause of this phenomenon we analyzed these single request sessions more rigorously. We found that for the 100 second timeout case, 50% of these single requests were for base files (e.g., HTML), 38% for embedded files (e.g., Image and Java), 6% for single files (e.g., Compressed) and 6% for non-cacheable responses (e.g., Dynamic requests, error messages). This is vastly different from the overall file type distribution reported in Table 5. We believe that caching, either at the client or within the network, is responsible for many of these short sessions. That is, many user requests are being served from caches so substantially fewer requests are reaching the Web site. Embedded files in particular are likely to be cached, which is why we see such a change in the file type distribution. The popularity of the World Cup site may have added to this phenomenon by increasing the probability that its files would be stored in shared caches throughout the Internet. However, we speculate that if the network caching architecture continues to grow more and more sessions may consist of only a single request (or a few requests). Wide spread adoption of Web cache consistency mechanisms, including those in HTTP/1.1 [19], could also reduce the number of requests per session.

Figure 16(b) shows the distribution of total response content bytes sent from the World Cup site to the client during a session. Not only does the rightmost curve (Bytes per Client) indicate the best possible use of a session for this workload, it also reveals the amount of content data that each client received for the entire monitoring period. For example, 15% of all clients received more than 1 MB (2^{20} bytes) of data. Looking at Figure 16(b) we can see that for small timeout values (e.g., 1 or 10 seconds) about 10% of all sessions transferred no content data. These sessions consisted primarily of Not Modified responses, another indication of caching at work. With a 100 second timeout 50% of all sessions transferred between 64 KB and 1 MB of content (2^{16} - 2^{20} bytes).

5.6.6 Inter-Session Times

Our next analysis of sessions studies the 'off-times' between successive sessions from the same client. We calculate the off-time from the moment a session times out until the arrival of the first request in the client's next session. By eliminating the timeout value from the inter-session time we can determine how long a server would have been required to maintain the session before receiving the next request from the client. The distribution for the time between the last request of session $\mathbf{s}_{\boldsymbol{i}}$ and the first request of session $\mathbf{s}_{\boldsymbol{i+1}}$ can be determined by shifting the curve to the right by the timeout value.

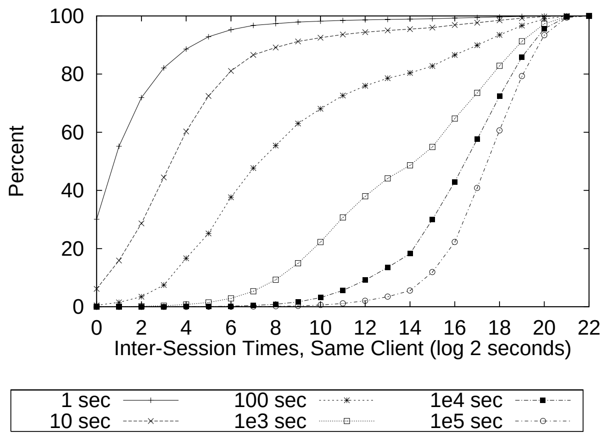


Figure 17 Analysis of Inter-Session Times

The results of this analysis are shown in Figure 17. For small timeout values the graph reveals that the sessions would have been reused had the server maintained them for a few additional seconds. For example, with a one second timeout more than half of the sessions could have been reused if the server had waited an additional two seconds before closing them. As the timeout values increase the server would need to maintain the sessions for a significantly longer period of time in order to see any further use. Assuming a 100,000 second timeout only 22% of the sessions could have been reused if the server had maintained them for an additional day (2^{16} seconds).

5.6.7 Intra-Session Times

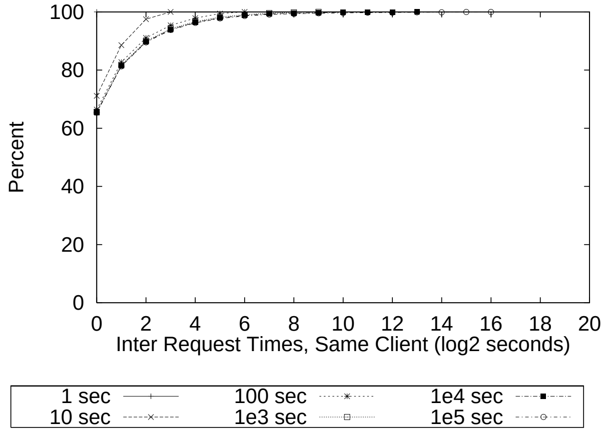


Figure 18 Analysis of Inter-Request Times in Individual User Sessions

Our final set of analyses in this section examine intra-session times. This information may be useful in developing more adaptive policies for managing TCP connections on a Web server.

We conducted two separate analyses. One of these analyses measured the time between requests in each distinct session. Figure 18 shows the cumulative frequency distribution for all of these inter-request times. Due to the coarse timestamp granularity, most of the inter-request times are either 0 or 1 second (over 60% for all session timeout values). This indicates that most of the requests in a user session are automatically generated by the client - i.e., the browser automatically retrieving all of the embedded objects in the Web page that the user requested. Most of the remaining inter-request times are less than 64 seconds (2^6). These correspond to the time between the last automatically generated request and the request for the next page that the user is interested in. In a few cases the inter-request time exceeds 64 seconds.

In order to get a better estimate of "user think times" (i.e., the time between a user requesting Web page ***i*** and Web page ***i+1***), we decided to monitor the time between requests for HTML objects in each distinct session. As expected, the inter-request times for HTML objects (shown in Figure 19(b)) are much longer than for all object types (Figure 18). There are fewer inter-request times of 0 or 1 second when only the HTML files are considered, due to fewer automatically generated requests. Since many of the World Cup Web pages utilized frames (i.e., were composed of several HTML objects) there are still a significant number of automatically generated requests. For the larger session timeouts (e.g., 1,000 to 100,000 seconds) approximately 45% of the inter-HTML request times are between 8 and 255 seconds (2^3 up to, but not including, 2^8) in duration. For these session timeout values Figure 19(a) indicates that the most common "use think times" are in the 32-63 second range (2^5 seconds). As the session timeout value increases we see a larger number of long inter-request times for HTML objects. While some of these are "user-think times", others result from the merging of multiple sessions into one logical session.

In our analyses a session ends when it has been "idle" for more than a threshold value (***t*** seconds). In other words the session will timeout when no request has been made by the client in more than ***t*** seconds. Using this definition no inter-request times greater than ***t*** will be seen. Thus, in Figure 18 all of the curves are bounded by the session timeout value. However, it is possible for the time between subsequent requests for HTML objects to exceed ***t***. For example, when HTML object ***i*** is requested, it is usually followed by a number of automatically generated requests for the embedded objects (e.g., the inline images). This process may take several (e.g., ***x***) seconds to complete, depending on the network connectivity, the server load, the number of embedded objects, etc. Following this there is typically an idle time (e.g., ***y*** seconds) as the user reads the Web page. The idle time ends when the user selects a hyperlink which results in the request of HTML object ***i+1***. If the idle time exceeds the timeout threshold (i.e., $y \geq t$) then the existing session ends and the request for the HTML object ***i+1*** starts a new session. If the idle time does not exceed the timeout threshold (i.e., $y < t$) then the existing session remains active and we calculate the inter HTML request time (***ihrt***) for objects ***i*** and ***i+1*** as ***ihrt=x+y***. For example, if ***x*=8**, ***y*=7** and ***t*=10**, then ***ihrt*=15**; this satisfies both the properties of $y < t$ and $ihrt > t$. Thus, it is possible for inter HTML request times to exceed the session timeout value. Therefore, the curves in Figure 19 are not bounded by the session timeout value.

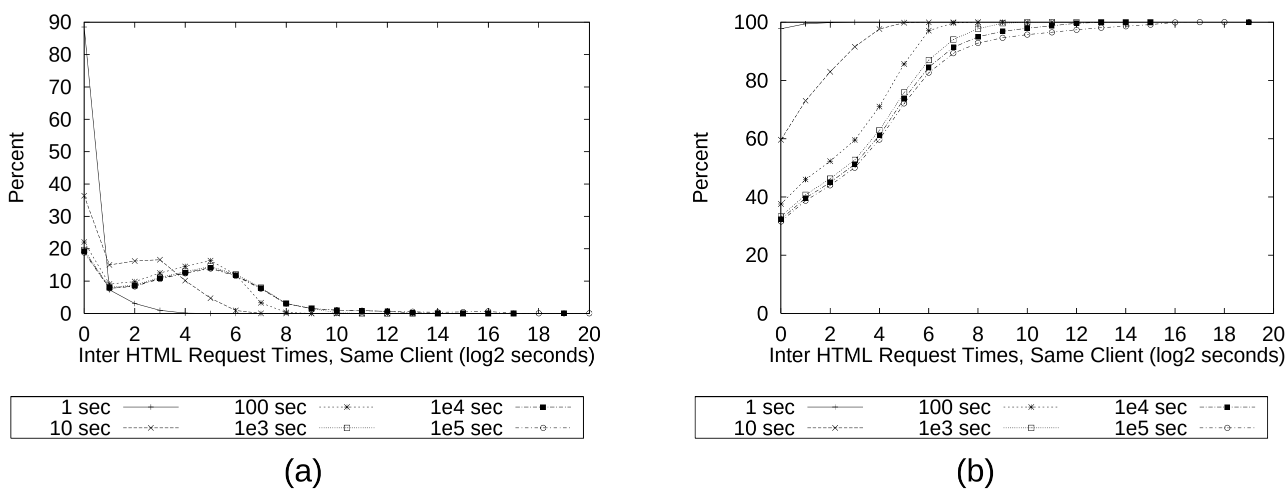


Figure 19 Analysis of Inter-Request Times for HTML Objects: (a) Frequency; (b) Cumulative Frequency

6 ANALYSIS OF A PEAK WORKLOAD

In Section 5 we characterized the World Cup workload across the entire data collection period. In Section 5.2 we noted that much of the traffic came in large bursts that occurred while football matches were in progress. In this section we analyze the workload from one of these large bursts and compare the results to those in Section 5. The purpose of this study is to determine what changes, if any, occur to the workload characteristics when the traffic is exceptionally heavy.

6.1 Analysis Period

For this analysis we chose the busiest 15 minute period from the overall World Cup workload. This period occurred from 11:30pm until 11:45pm, June 30th, 1998. During this time penalty kicks were being used to determine the victor in a playoff match between Argentina and England. For the remainder of this paper we shall refer to this subset of the overall World Cup workload as the A-E workload.

Table 11 Summary of Access Log Characteristics (A-E Workload)

Duration	11:30pm-11:45pm, June 30th, 1998
Total Requests	3,135,993
Avg Requests/Minute	209,066
Total Bytes Transferred (GB)	8.5
Avg Bytes Transferred/Minute (MB)	580

Table 11 reports some overall statistics on the A-E workload. Over three million requests were received by the World Cup site during the 15 minute period. The average number of requests received per minute was over 19 times the average rate for the overall workload (see Table 1 on page 8). The average rate of data transfer per minute for the A-E workload was 13 times that of the overall workload.

6.2 Statistical Characteristics

Table 12 Breakdown of HTTP Version (A-E Workload)

HTTP Version	% of Requests	% of Content Data Transferred
0.9	0.00	0.00
1.0	78.30	82.36
1.1	21.67	17.54
x.x	0.03	0.10
Total	**100.00**	**100.00**

Table 13 Breakdown of Resource Methods (A-E Workload)

Method	% of Requests	% of Content Data Transferred
GET	99.99	99.97
HEAD	0.01	0.03
POST	0.00	0.00
Total	**100.00**	**100.00**

Table 12 reports the breakdown of HTTP versions supported by clients in the A-E workload. The results are quite similar to those for the overall workload shown in Table 2 on page 11.

Table 13 lists the breakdown of resource methods for all of the requests in the A-E workload. This breakdown is also quite similar to the overall results (Table 3 on page 13). For all remaining analyses we focus exclusively on the requests which utilized the GET resource method.

Table 14 Breakdown of Server Response Codes (A-E Workload)

Response Code	% of Requests	% of Content Data Transferred
200 (Successful)	62.63	99.94
206 (Partial Content)	0.01	0.04
304 (Not Modified)	37.18	0.00
4xx (Client Error)	0.18	0.02
5xx (Server Error)	0.00	0.00
Other Codes	0.00	0.00
Total	**100.00**	**100.00**

Table 14 shows the breakdown of server response codes from the A-E workload. This breakdown is quite different from the overall distribution provided by Table 4 on page 13. The most significant change between the workloads is the percentage of Not Modified responses. In the A-E Workload over 37% of all server responses were Not Modified. This is twice the percentage seen in the overall workload. This characteristic indicates that many of the clients are simply performing consistency checks to ensure that the World Cup files that they have stored in their caches are still up-to-date. This is likely the result of users hitting the 'reload' button on their browsers to check whether there has been a change in the status of the match.

Table 15 presents the breakdown of the requests in the A-E workload by file type. There are

Table 15 Breakdown by File Type (A-E Workload)

File Type	% of Requests	% of Content Data Transferred
HTML	7.36	67.84
Images	90.96	26.08
Audio	0.00	0.09
Video	0.00	0.14
Compressed	0.00	1.13
Java	0.34	0.45
Dynamic	0.00	0.00
Other Types	1.34	4.27
Total	**100.00**	**100.00**

several differences between the file type distribution for this workload and the overall workload reported in Table 5 on page 15. For example, in the A-E workload HTML files are now the dominant source of the content data transferred. This occurs because the HTML files are being modified to reflect changes in the status of the match and thus must be served in their entirety. Images, which account for most of the requests, account for a much smaller percentage of the content data transferred as many responses are simply acknowledgments that the file has not been modified and thus contain no content data. Since most users appear to be interested primarily in the status of the match, the compressed files are even less popular than normal and therefore have less impact on the total content data transferred in the A-E workload than they did in the overall workload.

Table 16 shows how the requests in the A-E workload were distributed across the four

Table 16 Breakdown by Location (A-E Workload)

Location	% of Requests	% of Content Data Transferred
Santa Clara, CA	13.36	13.68
Plano, TX	46.99	45.53
Herndon, VA	34.61	36.11
Paris, FR	5.04	4.68
Total	**100.00**	**100.00**

server locations. These results indicate that the Paris location received a substantially smaller percentage of the requests in the A-E workload compared to the overall workload (see Table 6 on page 17). Meanwhile the Herndon site received a larger percentage of the requests. Assuming that the clients were sent to a ‘geographically close’ location (which is not always the case) we would expect to see this behaviour, as most European users would likely be watching the match on television (recall that the match is in the late evening for European fans).

Table 17 Breakdown of Clients (A-E Workload)

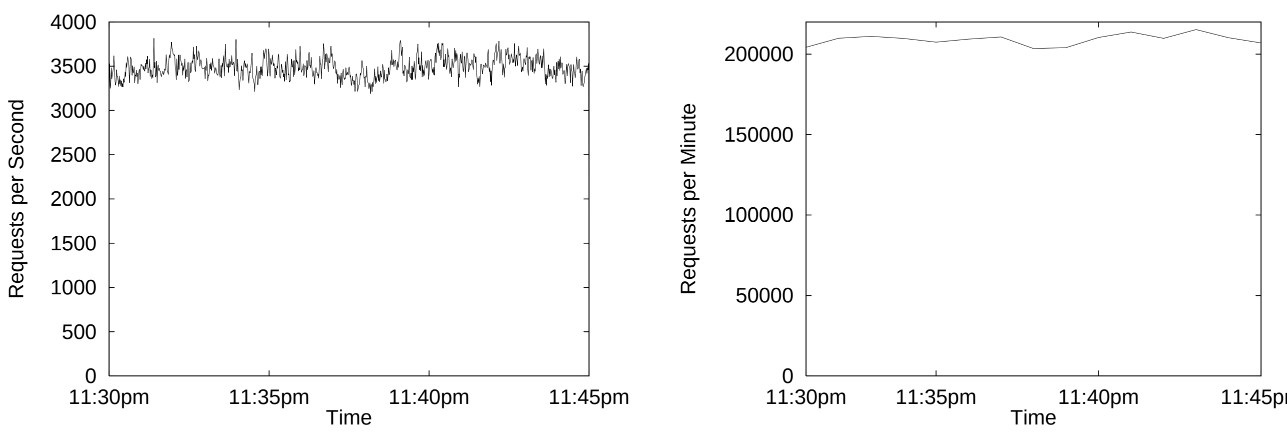
	Location[a]	% of Unique Clients	% of Requests	% of Bytes Transferred
Single Location	SC only	10.69	7.65	7.67
	PL only	35.61	32.62	30.96
	HN only	29.38	24.02	24.56
	PA only	7.04	3.52	3.25
Subtotal		82.72	67.81	66.44
Two Locations	SC & PL	4.55	7.16	7.44
	SC & HN	1.69	2.10	2.23
	SC & PA	0.20	0.20	0.20
	PL & HN	6.16	10.50	11.24
	PL & PA	1.02	1.26	1.31
	HN & PA	1.11	1.21	1.33
Subtotal		14.73	22.43	23.75
Three Locations	SC, PL & HN	1.35	5.46	5.50
	SC, PL & PA	0.22	0.77	0.76
	SC, HN & PA	0.13	0.21	0.25
	PL, HN & PA	0.55	1.55	1.57
Subtotal		2.25	7.99	8.08
Four Locations	SC, PL, HN & PA	0.30	1.77	1.73
Total		**100.00**	**100.00**	**100.00**

a. Abbreviation definitions are given in Table 21 on page 87.

Table 17 reports the breakdown of clients by the number of server locations that they contacted. A total of 20,531 unique clients were seen during the 15 minute A-E workload. Many of these clients (82.72%) contacted only a single location during this time. These clients accounted for 67.81% of all requests in the A-E workload and 66.44% of the bytes transferred. This should not be unexpected as the network dynamics should be relatively stable during this short period of time, although the ‘flash crowd’ could affect this. There were still a significant number of clients that contacted multiple locations although the percentages were much smaller than for the overall workload (Table 7 on page 19). There were even a few clients (0.30%) that sent requests to each of the four server locations during this 15 minute time frame. A lack of information prevents us from examining this in more depth.

6.3 Usage

Figure 20 shows the request rate for the A-E workload. Figure 20(a) reveals that throughout this 15 minute period the request rate is relatively stable, with an average rate of 3,484 requests per second and a peak rate of 3,816 requests per second. Figure 20(b) indicates that on a per minute basis the request rate is even more stable, peaking at 215,241 requests per minute and averaging 209,066 requests per minute.



(a) (b)

Figure 20 A-E Workload, Volume of Requests: (a) per second; (b) per minute

6.4 Size Distributions

6.4.1 Size Distribution of Unique Files

Table 18 Unique File Size Information by File Type (A-E Workload)

	All Files	HTML	Image	Audio	Video	Java	Compressed	Dynamic
Number	5,201	1,948	3,184	22	9	4	14	9
Mean (bytes)	15,238	11,418	6,054	375,163	1,458,673	4,043	1,139,356	33,510
Median (bytes)	4,850	6,066	3,876	139	1,367,199	4,406	1,419,393	25,596
Maximum (MB)	2.8	0.12	0.10	1.3	1.9	0.004	2.8	0.06
Total Size (MB)	75.6	21.2	18.4	7.9	12.5	0.02	15.2	0.29

Table 18 provides a breakdown of the size distributions by type for the unique files requested in the A-E workload. There are a number of differences compared to the unique size distribution of the overall workload (refer to Table 8 on page 26). For example, fewer unique files were accessed in the A-E workload. This indicates the focus of the users on a particular subject. Also, the number of unique HTML files accessed was substantially less, again indicating that the users were interested in a smaller set of the pages available at the World Cup site. Finally, fewer large files were accessed in the A-E workload.

Figure 21 presents the unique file size distribution for the A-E workload. The results show that the distributions are quite similar to those from the overall workload (see Figure 3 on page 27) except that no extremely large files (e.g., greater than 10 MB) were seen in the A-E workload.

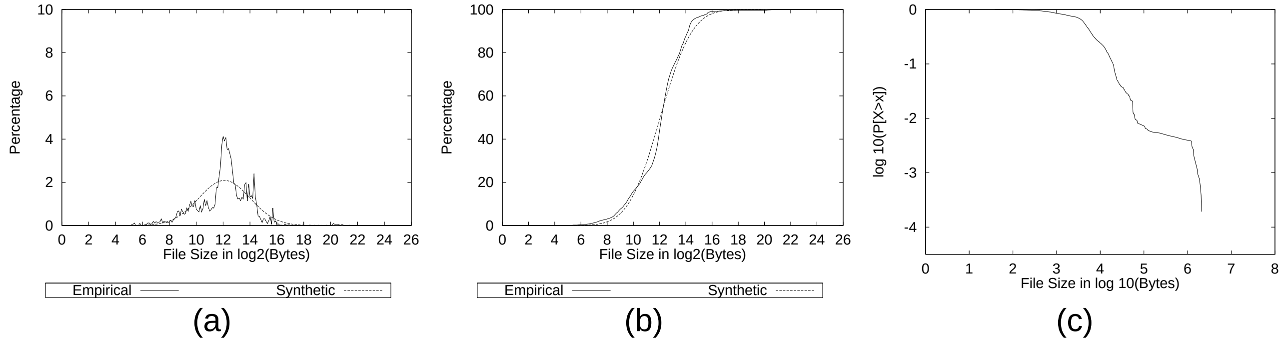


Figure 21 Size Distribution of Unique Files, A-E Workload: (a) Frequency; (b) Cumulative Frequency; (c) Tail

6.4.2 Size Distribution of Successful Transfers

Table 19 Successful Transfer Size Information by File Type (A-E Workload)

	All Transfers	HTML	Image	Audio	Video	Java	Compressed	Dynamic
Number	1,963,850	183,207	1,735,079	139	9	9,755	136	20
Mean (bytes)	4,619	33,609	1,363	59,486	1,458,673	4,186	738,062	17,884
Median (bytes)	933	46,941	872	124	1,367,199	4,406	263,198	6,218
Maximum (MB)	2.8	0.12	0.10	1.3	1.9	0.004	2.8	0.06
Bytes Transferred (GB)	8.5	5.7	2.2	0.01	0.01	0.038	0.09	0.0003

Table 19 breaks down the successful transfer size distribution by file type. The mean and median successful transfer sizes are quite similar to those reported in Table 9 on page 31 for the overall workload. For example the median successful transfer size in the A-E workload is 933 bytes compared to 965 bytes for the overall workload. Perhaps the most significant difference is the changes in the mean and median transfer sizes for HTML files. For example the median HTML transfer size nearly quadrupled to 46,941 bytes in the A-E workload from 12,624 bytes in the overall workload.

Figure 22 shows the graphs of the body and tail of the successful transfer sizes in the A-E workload. These distributions are similar to those for the overall workload shown in Figure 5 on page 32. The main difference is in the tail of the distributions, due to fewer large files being requested in the A-E workload.

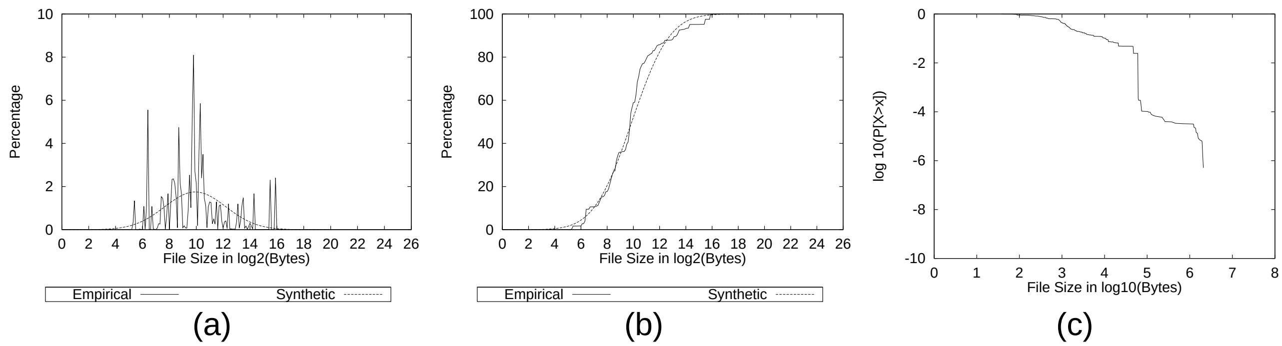


Figure 22 Size Distribution of Successful Transfers, A-E Workload: (a) Frequency; (b) Cumulative Frequency; (c) Tail

6.4.3 Size Distribution of All Transfers

Figure 23 shows the size distribution for all transfers in the A-E workload. The increase in Not Modified responses in this workload adds significantly more 0-sized transfers to Figure 23(a). The presence of these 0-sized responses lowers the median transfer size to 305 bytes compared to 828 bytes for the overall workload (described in Section 5.3.3). The tail of the transfer size distribution for the A-E workload, shown in Figure 23(b) is not as heavy as the overall workload (see Figure 6(b)), indicating a lower probability that large files will be requested.

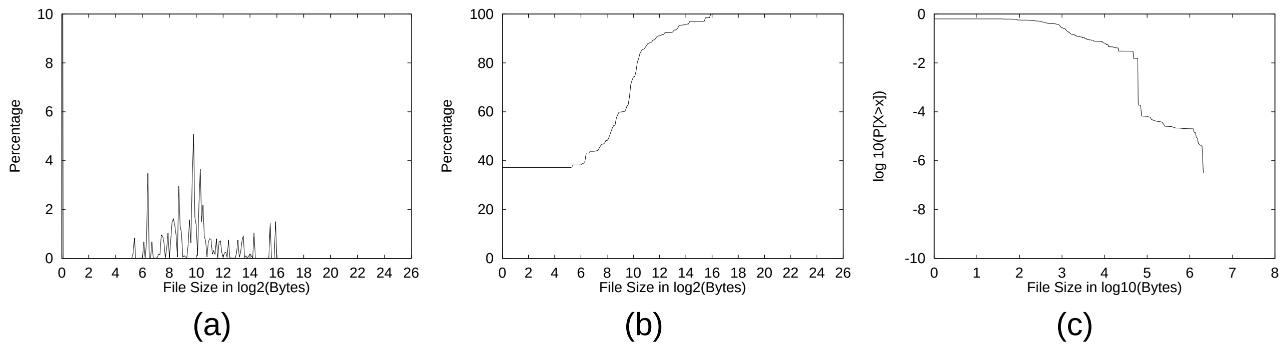


Figure 23 Size Distribution of All Transfers, A-E Workload: (a) Frequency; (b) Cumulative Frequency; (c) Tail

6.4.4 Impact of Size Distributions

Figure 24 relates the size of the unique files requested in the A-E workload to the number of requests and bytes transferred. As was the case in the overall workload (Figure 7 on page 34) most of the unique files are quite small while most of the storage space is consumed by a few large files. Also, most of the requests to the site are for the extremely small files. The one significant difference between the workloads is that in the A-E workload files in the 16-64 KB range account for most of the bytes transferred; larger files have little impact on the network bandwidth. In the overall workload responses containing files larger than 64 KB accounted for 21% of all bytes transferred.

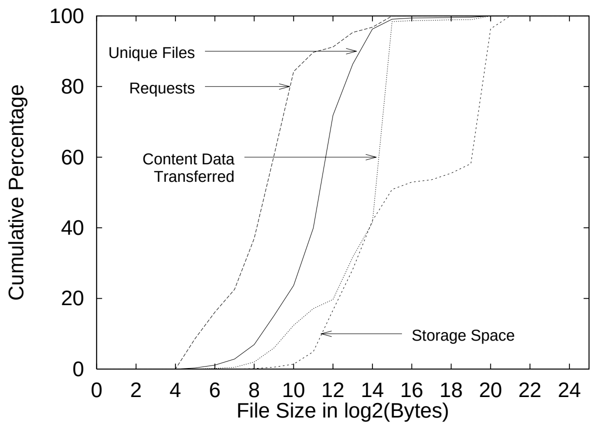


Figure 24 Impact of Size Distributions on Server and Network, A-E Workload

6.5 File Referencing Behaviour

6.5.1 Temporal Locality

Table 20 lists the results of the stack depth analysis for the A-E workload. Compared to the

Table 20 Temporal Locality Analysis (A-E Workload)

	All Locations	Santa Clara	Plano	Herndon	Paris
mean stack depth	75	62	71	73	85
standard deviation	147	72	123	111	123
median stack depth	52	49	51	53	59
90th percentile	121	111	116	125	154
normalized mean stack depth	0.014	0.012	0.014	0.014	0.016
normalize median stack depth	0.010	0.009	0.010	0.010	0.011

results in Table 10 on page 36 for the overall workload, both the mean and median stack depths are substantially shorter. This difference illustrates the interest in a smaller set of files at the World Cup site during the A-E workload.

Figure 25 provides the frequency, cumulative frequency and log-transformed cumulative frequency histograms for the stack depth distribution for the A-E workload. These graphs indicate that the temporal locality is much stronger in the A-E workload than it was in the overall workload. That is, the top of the stack received a much higher percentage of references in Figure 25 than in Figure 8 on page 37 (the overall workload).

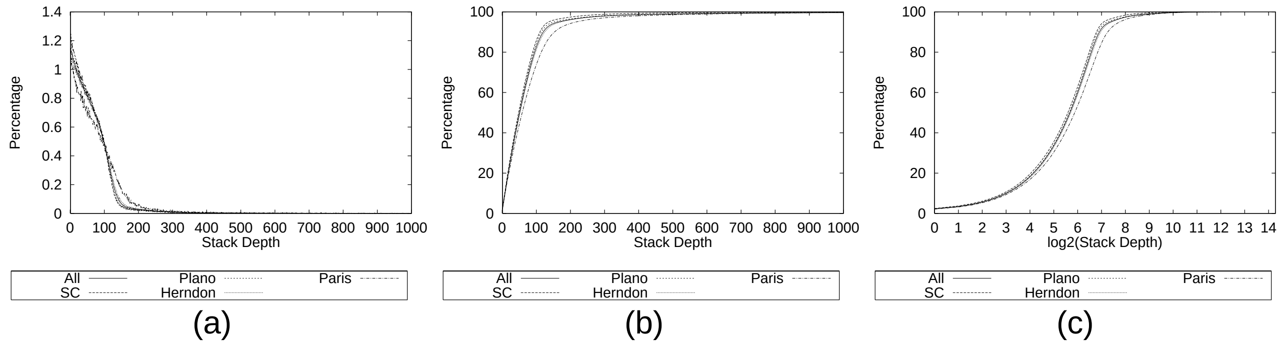


Figure 25 Stack Depth Distribution, A-E Workload: (a) Frequency; (b) Cumulative Frequency; (c) Log-Transformed CF

6.5.2 Concentration of References

Figure 26 shows the distribution of all client requests across the unique files in the A-E workload. The results in Figure 26 show that the references in the A-E workload were even more concentrated than they were in the overall workload (Figure 9 on page 38). For example, the most popular 10% of the unique files in the A-E workload received 99% of the requests and accounted for 96% of the content data transferred while occupying only 2% of the total storage space. In the overall workload the most popular 10% of files accounted for only 97% of references and 89% of the bytes transferred.

One-timers are much more prevalent in the A-E workload than they were in the overall workload. 2,069 (39.8%) of the 5,201 unique files in the A-E workload were accessed only a single time. These files accounted for 59.0% of the total size of the unique files accessed during this collection period. These observations indicate that very few people were "browsing" through the site during this period; the attention of most users was on a few extremely popular pages.

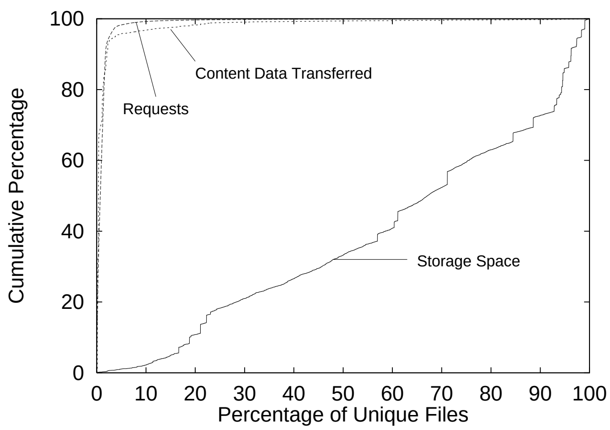


Figure 26 Concentration of References, cumulative distribution, A-E Workload

Figure 27(a) shows the relative popularity of the unique files referenced in the A-E workload. As was the case with the overall workload (Figure 10 on page 40) the popularity ranking does not appear to follow a Zipf-like distribution. In Figure 27(a) two linear regions are evident; the first for files 1-100 and the second for files 100-5,201. Figure 27(a) does not have the third linear region that is present in Figure 10(a) for the overall workload.

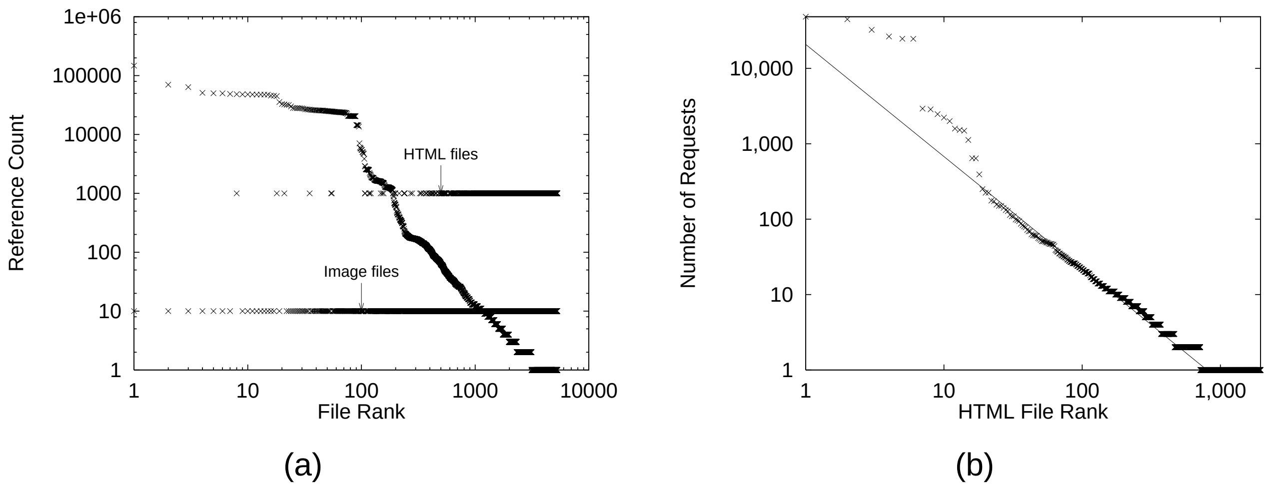


Figure 27 Concentration of References, reference count vs. rank, A-E Workload: (a) All Files; (b) HTML files only

Figure 27(b) presents the relative popularity of the HTML files referenced in the A-E workload. Two distinct regions can be seen in this graph. In region I (files 1-20) the files are substantially more popular than the files in region II (files 20-1,948). In region II the graph is roughly linear with slope estimated at -1.5. Thus a Zipf-like distribution would not accurately capture the concentration of references to the most popular files. A Zipf-like distribution may still provide a reasonable approximation for some testing purposes.

6.6 User Session Analyses

In this section we analyze the user sessions from the A-E workload using the approach described in Section 5.6. We then compare the results for this workload with those for the overall workload.

6.6.1 Total Sessions

Figure 28 shows the analysis results for the number of user sessions calculated for various timeout values. Since the A-E workload is only 900 seconds in duration we only tested three timeout values: 1, 10, and 100 seconds. With this workload the two extreme cases are 3,135,993 sessions when each HTTP request utilizes its own TCP connection, and 20,531 sessions when each unique client in the workload receives a persistent connection to use for the duration of the analysis period.

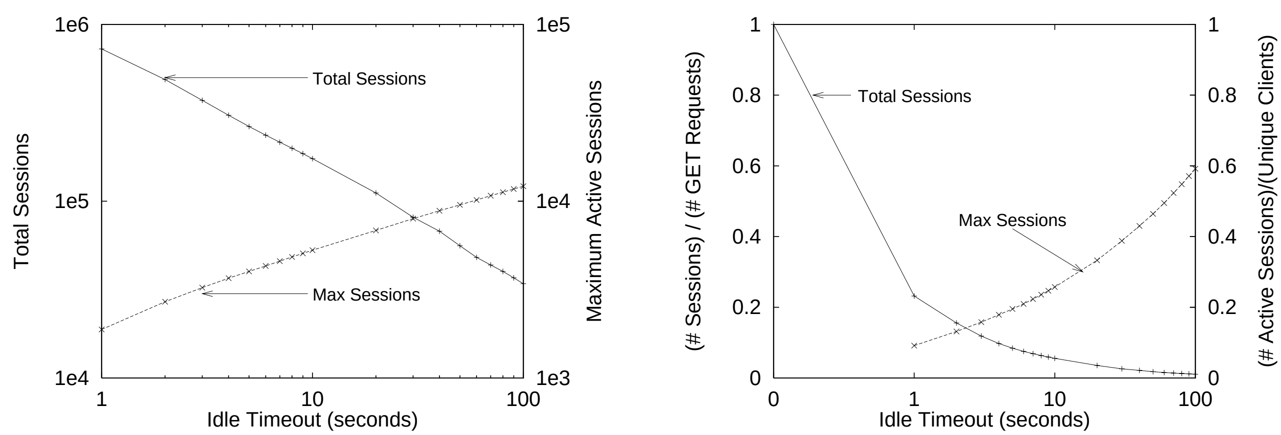


Figure 28 Effect of Timeout Values on Total Number of Sessions (A-E Workload)

Figure 28(a) shows the total number of sessions seen for each of the timeout values examined along with the corresponding maximum number of active sessions. For example, with a 10 second timeout, a total of 174,123 sessions occurred with at most 5,282 active at once. As expected, the total number of sessions decreases and the maximum number of active sessions increases as the timeout value increases.

Figure 28(b) compares the total number of sessions and the maximum number of active sessions to the extreme cases. For example, when a 10 second timeout is used only 5.6% of the total sessions occur compared to when each HTTP request uses its own TCP connection. At the same time 25.7% of the unique clients have active sessions. The main difference between Figure 28(b) and Figure 12(b) in section 5.6.1 on page 47 is the fraction of clients that have an active session. There is also a slightly better reuse of sessions in the A-E workload as indicated by the lower percentage of total sessions to GET requests for equivalent timeout values.

6.6.2 Active Sessions

Figure 29 shows the number of active sessions measured at the start of each one second interval. Figure 29(a) reports the results for a one second session timeout value. With this timeout value the number of active sessions is quite variable. This variability is related to

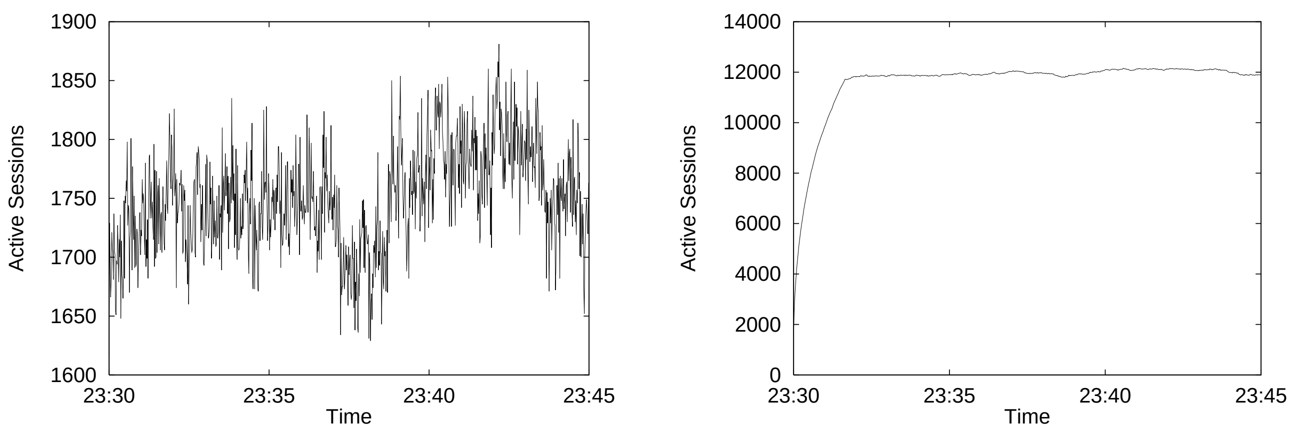


Figure 29 Active Sessions Over Time, A-E Workload: (a) 1 Second Timeout; (b) 100 Second Timeout

sessions timing out at the server before the client returns to the site to recheck the status of a match. When the client does return a new session must be created. Figure 29(b) provides the results for a 100 second session timeout value. In this graph the number of active sessions is quite stable. During the first two minutes we can see the growth in active sessions as more and more clients visit this site. This growth is mainly an artifact of our analysis having no knowledge of the active sessions prior to the start of the A-E workload. Once most of the unique clients (around 12,000) have established sessions with the site they appear to reuse their sessions within 100 seconds. This behaviour maintains their session for the duration of the workload.

6.6.3 Session Length

Figure 30 shows the length of sessions for the tested timeout values. In the A-E workload sessions tended to last slightly longer when short timeout values were used (e.g., 1 second) but not as long when greater timeout values were used (e.g., 100 seconds) compared to the overall workload (Figure 14 on page 51). Also, the session lengths in the A-E workload are constrained by the duration of the workload (15 minutes or 2^9 seconds).

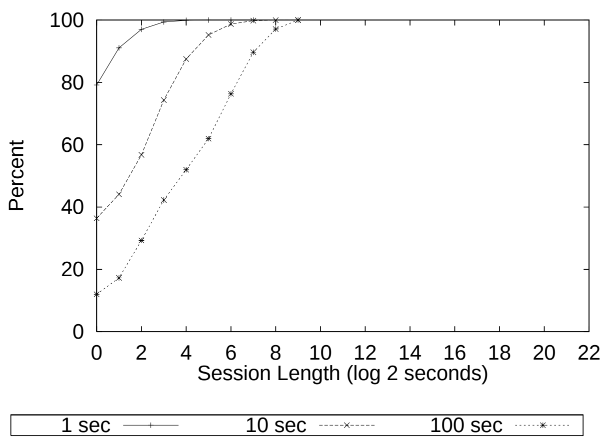


Figure 30 Analysis of Session Lengths, A-E Workload

6.6.4 Sessions Per Client

Figure 31 shows the number of sessions per client for the A-E workload. Due to the short duration of this workload the reduced number of sessions per client is to be expected when compared to the overall workload results (Figure 15 on page 52). however, Figure 31 indicates that many clients repeatedly visited the site during the 15 minute period that we analyzed. This behaviour is consistent with the reloading of a page to check on the status of the match.

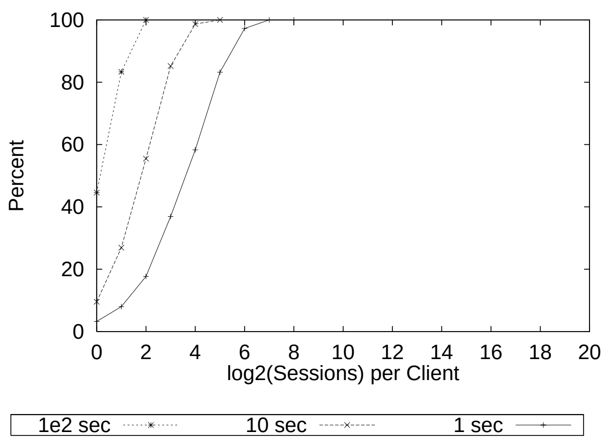


Figure 31 Analysis of the Number of Sessions Per Client, A-E Workload

6.6.5 Requests and Bytes Transferred Per Session

Figure 32 shows the number of requests made and the number of content bytes transferred per session for the A-E workload. Comparing these results to Figure 16 on page 53 for the overall workload results reveals that the distributions are quite similar. Figure 32 shows that fewer sessions are sending only a single request; that is, the sessions in the A-E workload were reused more often, particularly with the 100 second timeout. In Figure 32(b) more of the sessions (short timeouts only) transferred no content. This is due to the increased volume of cache consistency traffic.

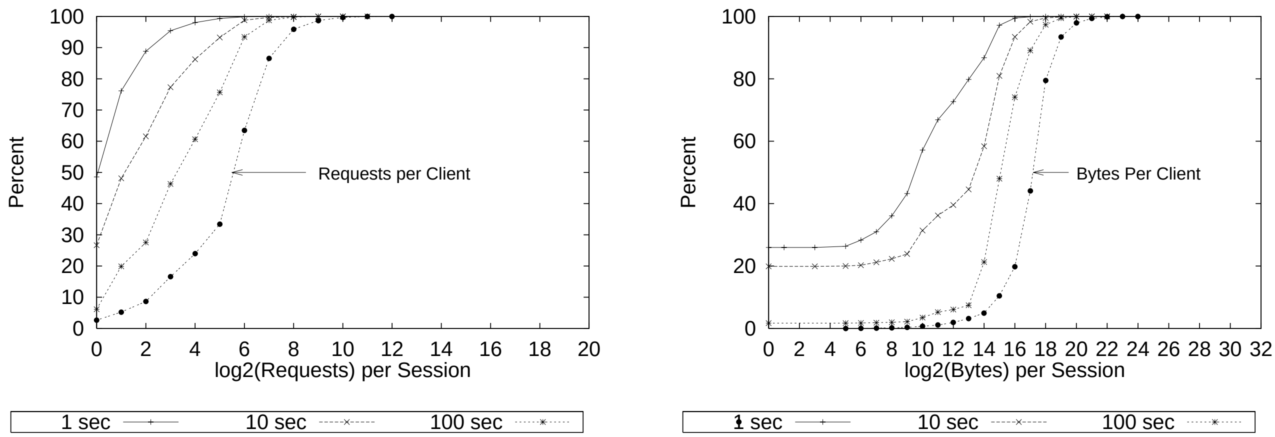


Figure 32 Analysis of Per Session Activity, A-E Workload: (a) Requests; (b) Bytes Transferred

6.6.6 Inter-Session Times

Figure 33 shows the inter-session time distributions for the A-E workload. There are two main differences between these results and those from the overall workload reported in Figure 17 on page 56. The first difference is that the tails in Figure 33 are much shorter. This characteristic is expected since the A-E workload is only 15 minutes (2^9 seconds) in duration. The other difference is the inter-session time distribution for the 100 second timeout. In the A-E workload the inter-session times for this timeout value are much shorter than in the overall workload. For example, about 55% of the sessions (assuming a 100 second timeout) in the A-E workload were reestablished 32 seconds or less after the previous session timed-out. In the overall workload only 17% of sessions were reestablished in this amount of time. This characteristic of the A-E workload indicates that many users were accessing the World Cup site about every two minutes (32 seconds + 100 seconds for the previous session to timeout) to check on the progress of the football match.

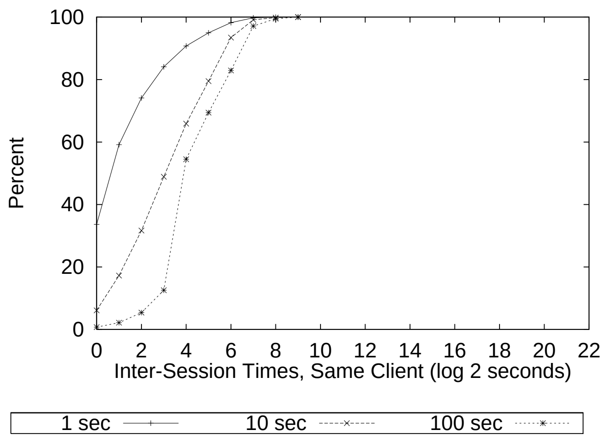


Figure 33 Analysis of Inter-Session Times, A-E Workload

6.6.7 Intra-Session Times

Our final analysis examines the intra-session times for the A-E workload. Figure 34 provides the cumulative frequency histogram of the inter-request times collected from each distinct session. This figure indicates that a higher percentage of requests appear to be machine-generated in the A-E workload than was the case in the overall workload (refer to Figure 18 on page 57). This observation is consistent with our hypothesis that most users in the A-E workload were simply reloading the same page again and again.

Figure 35 shows the inter-request time distribution for HTML objects. These results indicate that there were substantially fewer automatically generated HTML requests (as indicated by the 0-1 second spacing), while more of the inter-request spacings were in the 4-127 second (2^2 upto, but not including 2^7). These results seem consistent with users reloading a page multiple times over the duration of the workload.

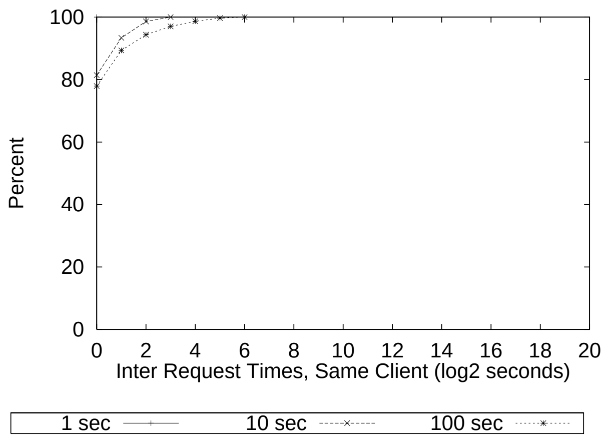


Figure 34 Analysis of Inter-Request Times in Individual User Sessions, A-E Workload

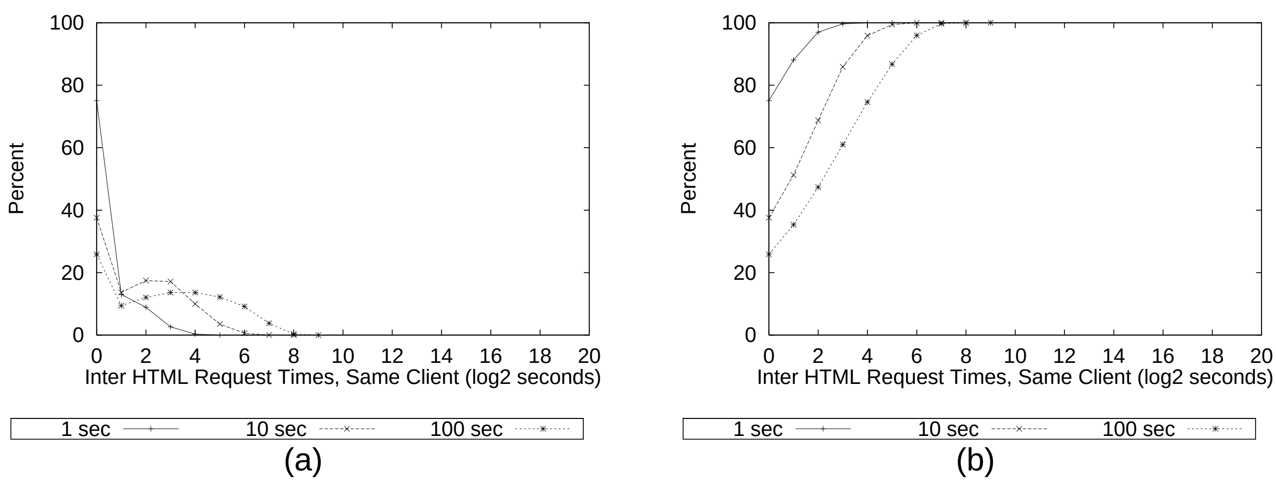


Figure 35 Analysis of Inter-Request Times for HTML Objects, A-E Workload: (a) Frequency; (b) Cumulative Frequency

7 PERFORMANCE IMPLICATIONS

During our workload characterization study (Section 5 and Section 6) we examined numerous characteristics of the World Cup workload. In this section we discuss the implications of several of these characteristics on Web server performance.

In Section 5.6.5 we discovered that a significant number of user sessions (17% when a 100 second timeout was used) contained only a single request during the lifetime of the session. There is no benefit in maintaining a persistent connection for this type of session, particularly for the server that must reserve resources for the connection. This characteristic of user sessions suggests that a trivial fixed length timeout policy for closing idle connections on the server may be inadequate. A more appropriate, but still relatively simple approach would be to utilize an adaptive timeout scheme like the one suggested by Mogul for dealing with proxies that do not support persistent connections [30]. With this approach the initial timeout value is quite small, so that if the connection is not reused it will quickly be considered idle and be closed by the server. If the connection is reused the timeout value would be increased to a more appropriate value. More adaptive TCP connection management policies for Web servers may also be useful. For example, a Web server could automatically adjust the idle timeout value in order to keep the number of active sessions within a specified range. A number of TCP connection management policies for persistent HTTP have been examined by Cohen *et. al.* [9].

Our previous work on Web server workload characterization [3] analyzed access logs that predated the widespread use of browsers with persistent (i.e., disk) caches, proxy caches, and (transparent) network caches. Today, Web server workloads have changed due to the growth of a Web caching architecture. In particular, many Web server responses are ‘Not Modified’ and contain no content data. The results of this workload characterization study have identified several other ways in which caching is altering Web server workloads. In our analysis of user sessions in Section 5.6.5 we determined that caching, in some cases, is reducing the number of requests that might utilize a persistent connection. The implications of this characteristic on server design are discussed at the beginning of this section. Further research is required to determine if fewer requests per session is a growing trend, and if so, what are the implications on Web servers and on HTTP.

A second observation regarding the effects of caching on Web server workloads was made in Section 6. From the perspective of Web server performance the main benefit of client, proxy and network caching is the reduction in workload at the server, particularly during periods of extreme user interest. Our results in Section 6 indicate that the lack of an efficient consistency mechanism is preventing Web servers from fully benefiting from caching. In other words when portions of the Web site’s content is extremely popular the site’s servers are not seeing a substantial reduction in workload. Instead of responding to a large number of GET requests, the servers must respond to a large number of cache consistency requests (i.e., GET If Modified Since requests).

Many of the requests for consistency information in the World Cup workload were caused by the caching of static image files. Much of this traffic could have been eliminated if the consistency functionality of HTTP/1.1 [19] had been utilized by the site and supported by the caches. However, the consistency mechanisms in HTTP/1.1 are not adequate in all situations (e.g., the modification patterns of some files are unpredictable, like the score of a football match that is in progress). Furthermore, it remains to be seen whether this functionality of HTTP/1.1 will meet the needs of content providers or whether a new, more automated system will be required. Several research efforts have looked at alternative cache consistency mechanisms [16][25][26]. If a new system is indeed required it should include a method of propagating only the changes to the cache storing the old version of a file, as suggested by Mogul *et. al.* [29].

During the World Cup tournament an estimated 13 million cumulative users visited the France '98 Web site. During this same period an estimated cumulative audience of 40 billion watched the matches on television. While the gap between Internet users and television audiences is partially due to restricted Internet access in some countries the main reason is clear - the audiences preferred live (high quality) video to still images and text descriptions. This suggests the integration of video and the Web may be necessary to reach a much greater portion of the world's population. Adding high quality video to a Web site would have significant performance implications.

8 Summary, Contributions and Future Work

This paper has presented a detailed workload characterization study of the 1998 World Cup Web site. The data set analyzed in this study contained 1.35 billion requests collected over a three month period, making this the largest Web server characterization study to date.

Throughout the paper emphasis was placed on comparing the characteristics of the World Cup workload to those observed in other Web server workloads.

The results of our study revealed that caching at Web clients, proxies and within the network is changing the workloads seen by Web servers. The lack of an efficient, supported and widely adopted cache consistency mechanism is the main cause of these changes and the primary reason why Web caches are failing to significantly reduce Web server workloads during times of extreme user interest in the content on those servers.

This paper presented preliminary results on many different facets on the workload of the World Cup Web site. Further, more in-depth analyses are needed on many of the topics discussed in this paper. For example, more precise modeling of sessions is required to evaluate the effects of longer sessions on server resource utilization. Other future work in this area includes developing new or reconfiguring existing Web server benchmarks to reflect current workloads. Such benchmarks are needed to more accurately estimate the performance of a particular server configuration. Additional workload characterization is needed, particularly of sites on an ongoing basis, to determine if the characteristics observed in this data set are present in others and to understand how these characteristics change over time. In order to perform more accurate analyses in the future, more precise measurements of (server) workloads are needed. This may involve changing the data collected in access logs (e.g., store finer-grained timestamps) or utilizing alternative methods of data collection (e.g., system instrumentation). Finally, as we alluded to earlier in this paper, a more efficient cache consistency mechanism, preferably one that requires little human intervention, is needed to further the scalability of the Web.

9 ACKNOWLEDGMENTS

The authors would like to thank all of the people who made this work possible. The authors are particularly grateful to the people at EDS who provided the World Cup access logs; to Christian Hostelet, HP Technical Director of the World Cup as well as Joel Dubedat and Jean Le Saint for providing information on the World Cup Web site architecture; to Katey Kennedy and Robert Slinn for providing statistics on the television audiences for the World Cup; to Mark Crovella of Boston University, for his assistance with the statistical analyses; and to Paul Barford of Boston University, Jim Pitkow of Xerox PARC, and Sharad Singal and Gary Herman of HP Labs for their constructive comments on the paper.

10 APPENDIX A

Table 21 Abbreviations for Hosting Locations

Abbreviation	Location
SC	Santa Clara, CA
PL	Plano, TX
HN	Herndon, VA
PA	Paris, France

Table 22 Abbreviations for Team Names

Abbreviation	Team	Abbreviation	Team
ARG	Argentina	ITA	Italy
AUT	Austria	JAM	Jamaica
BEL	Belgium	JPN	Japan
BGR	Bulgaria	KOR	South Korea
BRA	Brazil	KSA	Saudi Arabia
CHI	Chile	MEX	Mexico
CMR	Cameroon	MOR	Morocco
COL	Columbia	NGA	Nigeria
DEN	Denmark	NOR	Norway
ENG	England	PAR	Paraguay
ESP	Spain	ROM	Romania
FRA	France	RSA	South Africa
GER	Germany	SCO	Scotland
HOL	The Netherlands	TUN	Tunisia
HRV	Croatia	USA	United States
IRN	Iran	YUG	Yugoslavia

11 References

[1] V. Almeida, A. Bestavros, M. Crovella and A. de Oliveira, “Characterizing Reference Locality in the WWW”, Proceedings of *1996 International Conference on Parallel and Distributed Information Systems (PDIS ‘96),* pp. 92-103, December 1996.

[2] M. Arlitt, R. Friedrich and T. Jin, “Workload Characterization of a Web Proxy in a Cable Modem Environment”, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 27, No. 2, pp. 25-36, August 1999.

[3] M. Arlitt and C. Williamson, “Internet Web Servers: Workload Characterization and Performance Implications”, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 5, pp. 631-645, October 1997.

[4] P. Barford and M. Crovella, “A Performance Evaluation of HyperText Transfer Protocols”, *Proceedings of ACM SIGMETRICS ‘99*, Atlanta, GA, pp. 188-197, May 1999.

[5] P. Barford, A. Bestavros, A. Bradley and M. Crovella, “Changes in Web Client Access Patterns”, to appear in *World Wide Web Journal*, Special Issue on Characterization and Performance Evaluation, 1999.

[6] P. Barford and M. Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, *Proceedings of ACM SIGMETRICS ’98*, Madison, WI, pp. 151-160, June 1998.

[7] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, “Web Caching and Zipf-Like Distributions: Evidence and Implications”, *Proceedings of IEEE Infocom ‘99*, New York, NY, March 1999.

[8] P. Cao and S. Irani, “Cost-Aware Proxy Caching Algorithms”, *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, Monterey, CA, pp. 193-206, December 1997.

[9] E. Cohen, H. Kaplan and J. Oldham, “Managing TCP Connections under Persistent HTTP”, *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada, May 1999.

[10] M. Crovella, personal communication (email), January 1999.

[11] M. Crovella and M. Taqqu, “Estimating the Heavy Tail index from Scaling Properties”, t*o appear in Methodology and Computing in Applied Probability*, Vol. 1, November 1, 1999.

[12] M. Crovella and A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, pp. 835-846, December 1997.

[13] C. Cunha, A. Bestavros, and M. Crovella, “Characteristics of WWW Client-Based Traces”, Technical Report TR-95-010, Boston University Department of Computer Science, April 1995.

[14] K. Delgadillo, “Cisco DistributedDirector”, Cisco White Paper, 1997. Available at `www.cisco.com`.

[15] J. Dilley, “The Effect of Consistency on Cache Response Time”, Hewlett-Packard Laboratories Technical Report HPL-1999-XXX, September 1999.

[16] J. Dilley, M. Arlitt, S. Perret and T. Jin, “The Distributed Object Consistency Protocol”, Hewlett-Packard Laboratories Technical Report HPL-1999-XXX, September 1999.

[17] B. Duska, D. Marwood and M. Feeley, “The Measured Access Characteristics of World-Wide Web Client Proxy Caches”, *Proceedings of USENIX Symposium of Internet Technologies and Systems (USITS)*, Monterey, CA, pp. 23-35, December 1997.

[18] A. Feldmann, R. Caceres, F. Douglis, G. Glass and M. Rabinovich, “Performance of Web Proxy Caching in Heterogeneous Bandwidth Environments”, *Proceedings of IEEE Infocomm ‘99,* New York, NY, pp.107-116, March 1999.

[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk-Nielsen, L. Masinter, P. Leach, and T. Berners-Lee, “RFC 2616 - Hypertext Transfer Protocol - - HTTP/1.1”, June 1999.

[20] FIFA Web site, `http://www.fifa.com/`

[21] H. Frystyk-Nielsen, J. Gettys, A. Baird-Smith, E. Prud’hommeaux, H. Wium-Lie and C. Lilley, “Network Performance Effects of HTTP/1.1, CSS1 and PNG”, *Proceedings of ACM SIGCOMM ‘97*, Cannes, France, September 1997.

[22] S. Gribble and E. Brewer, “System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace”, *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, Monterey, CA, pp. 207-218, December 1997.

[23] B. Krishnamurthy and M. Arlitt, “PROCOW: Protocol Compliance on the Web”, Hewlett-Packard Laboratories Technical Report HPL-1999-99, September 1999.

[24] B. Krishnamurthy, J. Mogul and D. Kristol, “Key Differences between HTTP/1.0 and HTTP/1.1”, *Proceedings of the Eighth International World Wide Web Conference*, Toronto, Canada, May 1999.

[25] B. Krishnamurthy and C. Wills, “Study of Piggyback Cache Validation for Proxy Caches in the World-Wide Web”, *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS),* Monterey, CA, pp. 1-12, December 1997.

[26] C. Liu and P. Cao, “Maintaining Strong Cache Consistency in the World-Wide Web”, *Proceedings of the 17th IEEE International Conference on Distributed Computing Systems*, May 1997.

[27] A. Mahanti and C. Williamson, “Web Proxy Workload Characterization”, Technical Report, Department of Computer Science, University of Saskatchewan, February 1999.

[28] S. Manley and M. Seltzer, “Web Facts and Fantasy”, *Proceedings of USENIX Symposium on Internet Technologies and Systems (USITS)*, Monterey, CA, pp. 125-133, December 1997.

[29] J. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy, “Potential Benefits of Delta Encoding and Data Compression for HTTP”, *Proceedings of ACM SIGCOMM ‘97*, Cannes, France, pp. 181-194, September 1997.

[30] J. Mogul, “The Case for Persistent-Connection HTTP”, *Proceedings of ACM SIGCOMM ‘95*, Cambridge, MA, pp. 299-313, 1995.

[31] Netscape Navigator Remote Control, `http://home.netscape.com/newsref/std/x-remote.html`.

[32] V. Paxson, “Empirically-Derived Analytic Models of Wide-Area TCP Connections”, I*EEE/ACM Transactions on Networking*, Vol. 2, No. 4, pp. 316-336, August 1994.

[33] C. Roadknight, I. Marshall and D. Vearer, “File Popularity Characterisation”, *Proceedings of the 2nd Workshop on Internet Server Performance (WISP ‘99)*, Atlanta, GA, May 1999.

[34] K. Thompson, G. Miller and R. Wilder, “Wide-Area Internet Traffic Patterns and Characteristics”, *IEEE Network*, Vol. 11, pp. 10-23, November/December 1997.

[35]W3C Web page, “Logging in W3C httpd”, `http://www.w3.org/Daemon/User/Config/Logging.html`

[36]World Cup Press Literature Kit.