Hierarchical Workload Characterization for a Busy Web Server

Adeniyi Oke [1], Rick Bunt

Department of Computer Science, University of Saskatchewan, Canada S7N 5A9

Abstract

This paper introduces the concept of a Web server access hierarchy—a three-tier hierarchy that describes the traffic to a Web server in three levels: as aggregate traffic from multiple clients, as traffic from individual clients, and as traffic within sessions of individual clients. A detailed workload characterization study was undertaken of the Web server access hierarchy of a busy commercial server using an access log of 80 million requests captured over seven days of observation. The behavioural characteristics that emerge from this study show different features at each level and suggest effective strategies for managing resources at busy Internet Web servers.

Key words: Workload characterization, Web resource management, Web caching

1 Introduction

The phenomenal growth of the World Wide Web continues to challenge network infrastructure and systems. Web servers are over-loaded with requests, networks become congested with Web traffic, and end-users experience poor response times. This continuing growth in the demand for Web-based services has motivated a great deal of research on how to improve the performance and scalability of the Web. Various research studies have been conducted from the client perspective [7,9,11] and the proxy perspective [13,15] to achieve these goals, but a common theme is to manage the resources of the Web server effectively [4–6,8,16]. Under proper management, request throughput and data

Email addresses: adeniyi.oke@sasktel.sk.ca (Adeniyi Oke), rick@usask.sk.ca (Rick Bunt).

[1] Present address: Saskatchewan Telecommunications, 2121 Saskatchewan Drive, Regina, SK S4P 3Y2, Canada.

throughput at the server can be increased and network bandwidth can be conserved.

The focus of this paper is the consideration of the different levels through which Web servers can receive requests from prospective clients: as requests from multiple clients, as requests from individual clients, and as requests within sessions from single clients. Our research seeks to shed light on two important questions: how do the characteristics of Web server workload change over the levels of the access hierarchy, and what are the implications of these characteristics for Web server resource management. We believe that the contributions from this study will enable Web server designers to gain insight into the design and provisioning of Web services and suggest ways of improving the effectiveness of Web protocols (HTTP/1.1). Important conclusions of this paper are:

- While the behavioural characteristics of aggregate server workload are generally consistent with other studies [5,6], there are a few important changes. We found that these changes are due not to the effect of upstream caching (as we anticipated), but rather to some specific features of the server workload studied.
- Two distinct types of clients, "human" clients and "non-human" clients, were identified at the server, and the behavioural characteristics of their reference patterns are dramatically different. This has significant implications for performance.
- A client-based resource management policy could be effective at the Web server if the reference pattern of "non-human" clients can be managed separately from the reference pattern of "human" clients.

The remainder of the paper is organized as follows. Section 2 presents the background and related work for the study. Section 3 describes the Web server workload characterized. Section 4 summarizes the characteristics of the Web server workload as it relates to the access hierarchy. Section 5 concludes the paper.

2 Background and Related Work

The performance and scalability of the World Wide Web depend on the combined effectiveness of resource management strategies at the client, in the network and at the server. Workload characterization is important to the understanding of the behavioural characteristics of Web traffic. Many workload characterization studies have been carried out at both the client side and the server side. These studies focused primarily on aggregate workloads from multiple users, and are useful in designing non-differentiated client policies for the provisioning of document services or to determine the effectiveness of Web

protocols. Several of these related studies are briefly discussed to illustrate the work that has been done in this area.

Catledge and Pitkow [9] first characterized the Web traffic of user sessions at the client side. Three categories of user reference patterns were identified—serendipitous (random), browsing (looking for items of interest) and searching (investigating specific topics). In a much larger study of Web user sessions, Cunha *et al.* [11] identified two types of reference patterns, described as mostly surfing and mostly working. Hine *et al.* [12] observed three user reference patterns referred to as wanderer, resident and sojourner. Barford *et al.* [7] examined whether or not the user reference patterns identified in [11] change over time and the implications of these changes with regards to Web resource management at the client side.

At the server side, Arlitt and Williamson [5] identified ten invariant characteristics of Web server workloads for effective resource management. Arlitt and Jin [6] subsequently analyzed the server workload of the 1998 World Cup Web site to explore the extent to which these ten invariant characteristics change over time. Although some of the invariant characteristics were found to have changed over time, the results supported the continued use of the resource management strategies suggested in [5]. Arlitt [4] further characterized an aggregate server workload by grouping it into Web user sessions at the server with a view to understanding the impact of the newly introduced HTTP/1.1 protocol on the resource management of Web servers. A hierarchical approach for characterization of Web server workloads collected at e-business sites and information provider sites was introduced in [2,3,16]. Menasce *et al.* [16] used a hierarchical approach to understand whether or not the characteristics and invariants identified in [5] are still valid for e-business workloads and Almeida *et al.* [2,3] applied this approach to identify, characterize and distinguish two major categories of search agents found in Web server workloads, namely *crawlers* and *shopbots*, and further assessed the impact of their reference patterns on caching performance.

In many respects, our work is similar to these related studies, but it differs in our motivation to understand how the characteristics of Web server workloads change as one moves through three levels of access aggregation. This hierarchical characterization enables Web server designers to gain clearer insight into resource management implications of the decisions they must make.

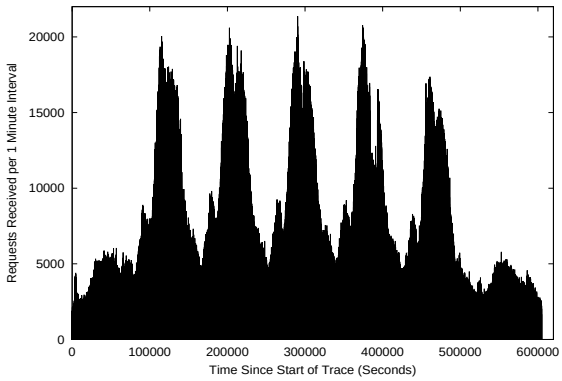


Fig. 1. Seven Days Traffic Profile of Web Server Workload Trace

3 The Collection, Description and Analysis of a Web Server Workload

For this study, we obtained Web server access logs from a busy (anonymous) commercial Web site. The Web site has a cluster of servers that appear to the users as a single machine. Each server has HP K-class multiprocessor HP-UX boxes running the Open-Market-Secure-Web Server/2.0.5.RCO. Server access logs were collected over a period of seven days from Sunday June 14, 1998 to Saturday June 20, 1998. The logs were preprocessed to preserve the anonymity of users and URLs. There are 644,187 clients [2] in the workload, and these are responsible for 328 GB of data transferred over the network within the period of observation. The daily server activity represents two orders of magnitude more workload than the busiest server activity characterized in [5] (tens of million of requests per day, compared to hundreds of thousands of requests per day). As Figure 1 illustrates, the original (or raw) workload shows a weekend effect with many fewer requests from clients during the weekend days than on work days.

After reducing the raw server access logs by eliminating unsuccessful requests (as was done in [5,6]), there are more than 56 million requests from 620 thousand unique clients over the period of observation. Table 1 gives summary statistics of the reduced access logs. In total, the server transferred more than 100 thousand unique documents over the seven-day period. The fact that the median requests per day is greater than the mean requests per day implies that the s weekend effect in the original workload is preserved in the reduced logs. There is evidence that the presence of a few large documents transferred by the server might be responsible for the skewness of the document size distribution since the median document size (9 kB) is less than the mean document

[2] The term client refers to a host with a distinct Internet Protocol (IP) address in the access log. A client can be a single user host, a timesharing cluster, or a relay host.

Table 1
Summary of Workload Statistics (reduced Workload)

Access Log Duration	7 days
Access Log Starting Date	June 14, 1998
Access Log End Date	June 20, 1998
Total Requests	56,458,479
Mean Requests/ Day	8,065,497
Median Requests/ Day	9,517,399
Coefficient of Variation	0.35
Total Bytes Transferred (GB)	323
Mean Transfer Size (Bytes)	5,722
Median Transfer Size (Bytes)	1,227
Coefficient of Variation	5.50
Total Unique Clients	620,041
Total Unique Documents	101,008
Total Storage Size (MB)	2,819
Mean Document Size (Bytes)	27,918
Median Document Size (Bytes)	8,506
Coefficient of Variation	9.74

size (28 kB). The mean transfer size (5.7 kB) and the median transfer size (1.2 kB) are consistent with results reported in [5,8]. One of our goals was to differentiate between documents that are heavily demanded and documents that are rarely or lightly demanded. By grouping the document types as reported in [5,6,15], some important differences are evident. In this workload, 9.51% of the requests were for HTML documents and 86.09% were for Image documents, in contrast with previous studies [5] which found these to be nearly the same. The volume of bytes transferred as HTML documents in our workload accounts for 23.66% of the traffic and the volume of bytes transferred as Image documents account for 63.28%. This indicates that Image documents have greater impact than HTML documents on this server, whereas the reverse was the case in [5]. The increasing popularity of Image documents over other types of Web documents could be responsible for this difference.

An important phenomenon is "one-timer" referencing with respect to documents and clients. One-timer documents are distinct documents transferred only once by the server during the period of observation. These are important because they adversely impact caching performance. In the reduced server

workload, there were 12,357 one-timer documents that represented 12.23% of the overall unique documents and 8.78% of total disk storage. One-timer clients are distinct users that successfully requested a document once from the Web server within the period of observation. These are also important because of their adverse impact on the effectiveness of persistent connections as implemented in HTTP/1.1. There were 33,205 one-timer clients in this workload, accounting for 5.35% of the number of clients. The large proportion of one-timer documents and one-timer clients in this workload suggests that caching strategies and the HTTP/1.1 protocol might not perform as hoped.

4 Hierarchical Workload Characterization

In characterizing the Web server workload a number of important characteristics were examined, including document size and transfer size distributions, temporal locality, document concentration, document inter-reference times and phase transition behaviour. These were studied at each of the aggregation levels described earlier.

4.1 Characteristics of the Aggregate Workload

The aggregate workload is simply the raw data presented to the server, with no attempt to isolate its component parts. Requests from the entire user community arrive at the server in an interleaved fashion.

4.1.1 Document and Transfer Size Distributions

The nature of documents either stored or transferred by the Web server is an important consideration for resource management. Figure 2(a) shows the cumulative frequency distribution of document references, document sizes and transfer sizes, and weighted transfer sizes [3]. Documents smaller than 10 kB account for 80.49% of the total requests and 28% of the total bytes transferred, documents between 10-100 kB account for 19.55% of the total requests and 65% of bytes transferred, and documents larger than 100 kB account for only 0.14% of the total requests and 7% of the bytes transferred. This indicates that Web users prefer smaller documents, probably less than 10 kB, while much of the data traffic on the network is due to a few requests for large documents. Caching should be an effective approach to resource management at

[3] Weighted transfer size is calculated by multiplying the number of references to a document by the maximum transfer size of the document within the period of observation.

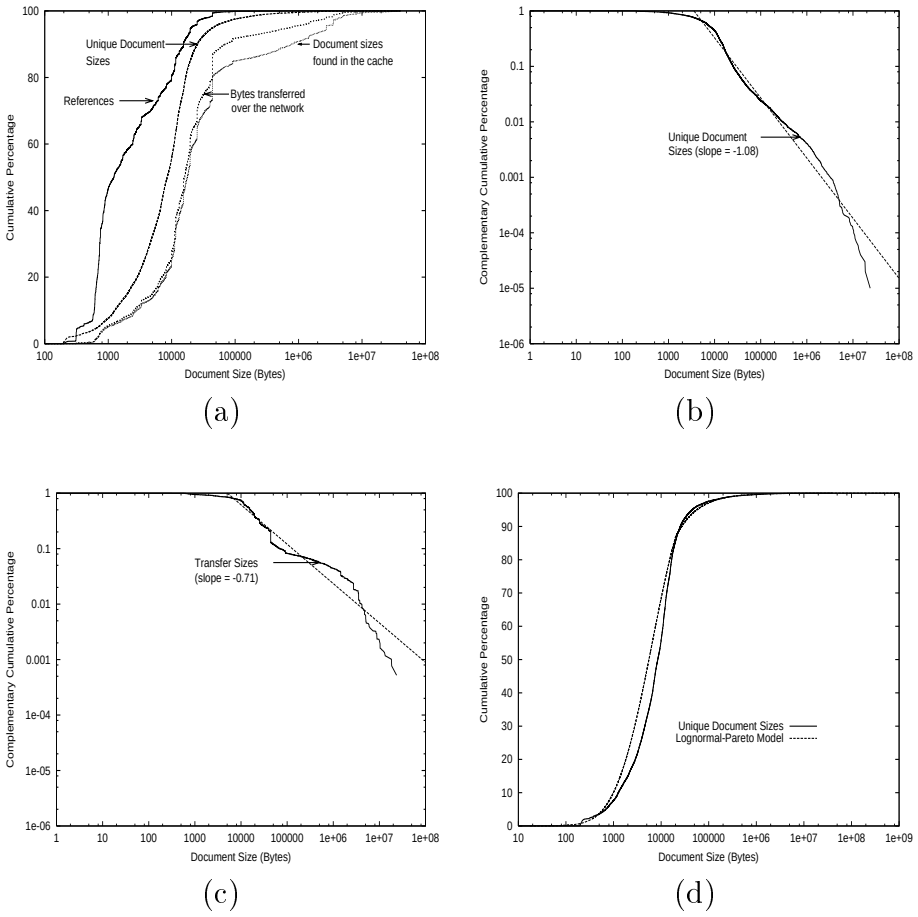


Fig. 2. Document Size Distribution: (a) Cumulative Distribution Frequency by Document Sizes; (b) Log-log Complementary Distribution Plot for Unique Documents; (c) Log-log Complementary Distribution Plot for Transfer Sizes; (d) Comparison of Document Size Distribution and a Hybrid Lognormal-Pareto model

the Web server, even though a tradeoff may exist between caching for improving the server's request throughput and caching for improving the server's data throughput, if cache policies are based on the frequency of user references to documents. The difference between the weighted transfer sizes and the actual transfer sizes for documents larger than 100 kB suggests that large documents are not fully downloaded before users abort the transfer. Thus, caching may be less effective for this category of large documents.

Figures 2(b) and (c) suggest that the distributions of document and transfer sizes are heavy-tailed or Pareto, respectively. For document sizes this means that a small set of large documents occupy most of the disk space, while for transfer sizes, a heavy tail means a small set of documents is responsible for most of the bytes transferred. Applying least-squares estimation (LSE) [5,6,10], the tail index of the document size distribution is $\alpha = -1.08$ (infinite variance) and the tail index of the transfer size distribution is $\alpha =$

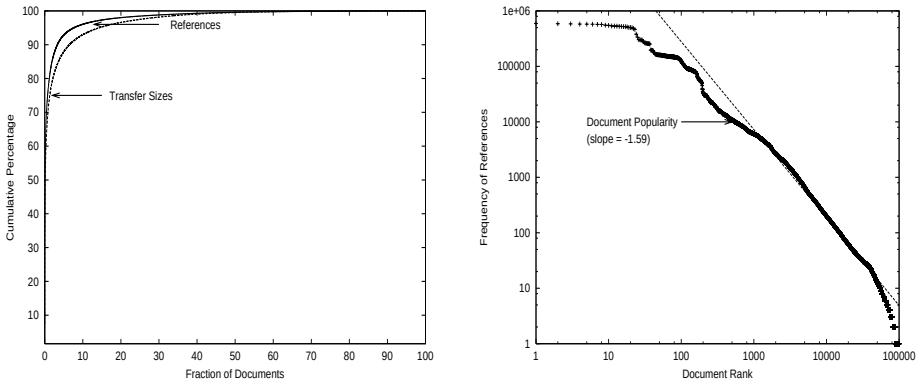


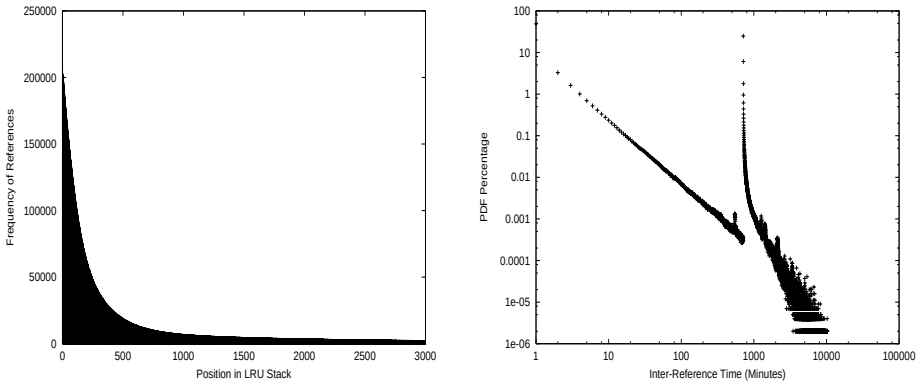
Fig. 3. Concentration of Document References: (a) Cumulative Distribution by References and Bytes Transferred; (b) Zipf Distribution: Reference Count versus Rank

-0.71 (infinite mean). These differ from earlier results [5,10,11] because of the clustering of the medium-sized image and text documents between 10 and 40 kB, that account for most of the bytes transferred. We also found that a hybrid lognormal-Pareto distribution as suggested by Barford *et al.* [7] captures the body and tail of the document distribution at the server well. Details are provided in [20].

4.1.2 Concentration of Document References

Several studies [5,6,11,15] have found that References to Web documents exhibit a non-uniform reference pattern. A small percentage of documents are extremely popular, while a significant number of documents are rarely referenced. Our server workload also exhibits this property as shown in Figure 3. Plotting distinct documents found in the server workload in decreasing order of reference (from the most popular to the least popular) against the fraction of total documents referenced shows that 10% of the most frequently accessed documents account for 96% of total requests and 5% of the most frequently referenced documents account for 93% of the overall requests received by the server (Figure 3(a)). Similar results are observed for bytes transferred by the server. The first 10% of distinct documents (representing the most heavily transferred documents) are responsible for 93% of bytes transferred over the network. Our results show the presence of strong document concentration in the aggregate workload and suggest that server caching strategies that take into account both the frequency of references and the volume of bytes contributed by documents can be effective.

Figure 3(b) demonstrates the use of Zipf's Law [23] to illustrate the presence of document concentration by plotting a log-log transformation of distinct documents, sorted in decreasing order of popularity, as a function of the doc-



(a) (b)

Fig. 4. Document Re-Referencing Behaviour: (a)Temporal Locality using LRU Stack Depth Histogram for 3000 Stack Positions; (b) Inter-Reference Times Probability Density Function Plot

ument rank. Although the slope of the popularity line is not -1.0, there is evidence of the document concentration property in this workload since the slope of the popularity curve is -1.59. The steeper the slope of the popularity line, the stronger the concentration or frequency of requests to a smaller set of documents. A similar observation was reported in [6] and suggests that caching holds promise as an approach to managing the Web server resources over time.

4.1.3 Temporal Locality

Temporal locality is essential for the success of caching strategies and is a significant factor in the choice of cache management policy. The least recently used (LRU) stack reference model [1,5,21] is used to characterize temporal locality since it reflects the tendency of references to certain documents to persist over time—the probability of users re-referencing documents that have been requested in the past. Figure 4(a) shows that temporal locality is present in this workload. The x-axis represents the first 3000 stack positions, from the most recently referenced document position to the least recently referenced document position, and the y-axis represents the frequency of reference to documents occupying each of the stack positions. The hyperbolic shape of the LRU stack distance histogram demonstrates that most documents occupy positions closer to the top of the stack, reflecting the presence of temporal locality in this aggregate workload.

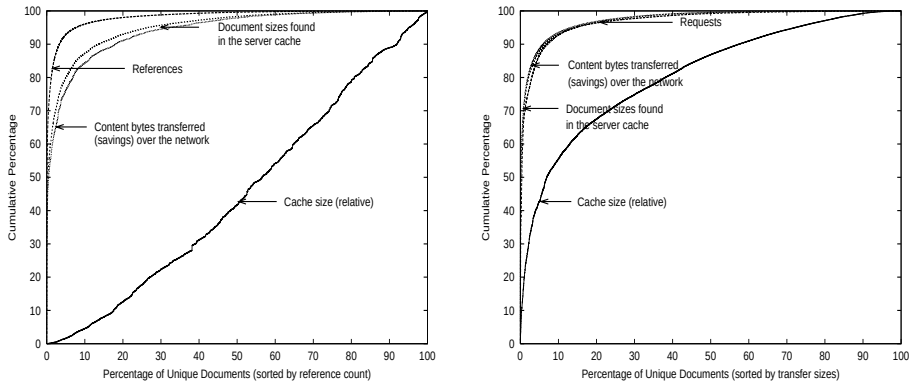


Fig. 5. Caching Implications for Web Server Resource Management

4.1.4 Re-referencing Behaviour: Document Inter-Reference Times

The degree of document re-referencing by users as exhibited by temporal locality is shown by the probability density function (PDF) of document inter-reference times (the time interval between successive requests to the same documents [13]). The intuition is that stronger (weaker) temporal locality reflects higher (lower) probability of smaller document inter-reference times. After computing and combining the inter-reference times of individual distinct documents in the workload, the PDF of overall document inter-reference times is shown in Figure 4(b). Since the probability of a user re-referencing the same documents decreases rapidly with time, the presence of temporal locality is concluded. This result is likely due to the presence of popular documents, which tend to be requested frequently, thus exhibiting shorter inter-reference times, while less popular documents exhibit longer inter-reference times. The break in the PDF of document inter-reference times at roughly 24 hour (1440 minute) intervals may be due to the impact of time zone differences in user referencing patterns, and the spike may be due to the presence of "non-human" references, such as might result from search agents or crawlers [2,3].

4.1.5 Resource Management Implications

There are two approaches to caching at a Web server: one is to cache documents that are frequently requested by users in order to improve the request throughput, the other is to cache documents that contribute to a large volume of bytes transferred in order to conserve network bandwidth or improve the data throughput. Figure 5 shows the implications of using cache policies based on the number of references to documents or on the volume of bytes contributed by documents on both the request throughput and the data throughput. The y-axis represents the cumulative percentage and the x-axis represents percentage of documents. Arlitt and Williamson [5] first used these

graphs in understanding the performance implications of Web servers.

Figure 5(a) depicts the performance results of using caching policies based on the number of references to documents. The top line represents the percentage of total requests serviced from the cache, the next two lines report the percentage of total bytes transferred (in terms of transfer sizes and weighted transfer sizes) for servicing requests from the cache, and the last line represents the cache size available at the server. The results show that there is a tradeoff in caching performance between request throughput and data throughput when the cache size increases. In particular, the cache policies perform better in terms of request throughput than data throughput. This implies that cache management strategies that are based on the frequency of requests to documents are likely to be effective for managing Web server resources when CPU cycles are the bottleneck.

Figure 5(b) shows the performance of cache management strategies based on the volume of bytes contributed by documents. The labeling of this graph is similar to Figure 5(a). In this case we observed no performance tradeoff between request throughput and data throughput as the cache size increases. This implies that cache management strategies based on volume of bytes contributed by documents are likely to be more effective when both CPU cycles and disk I/Os as are bottlenecks.

Frequency-based cache management strategies appear to be the most cost effective way of managing the resources at the Web server since the cost of a cache is always proportional to its size. Figures 5(a) and (b) show that in order to achieve comparable performance results, cache management strategies that use the number of references to documents require about 129 MB to store the top 10% of the most frequently requested documents, while cache management strategies that use the volume of bytes contributed by documents require about 1,564 MB to store the top 10% of documents that account for most volume of bytes transferred over the network.

4.2 Characteristics of Individual Client Workloads

For this part of our investigation, we decomposed the aggregate workload into separate streams comprising the requests from individual clients, and analyzed each of these client streams separately. In the design of resource management strategies, Web server designers assume (implicitly or explicitly) that client request patterns are similar. We found some dramatic differences that could have profound impact on performance. These are illustrated in this section by focusing on four specific clients that were identified.

Table 2
Summary of Server Client Access Log Statistics (Reduced Data)

Item Description	Client Number			
	A	B	C	D
Total Requests	109,385	97,859	21,122	9,594
Mean Requests/Day	15,626	13,980	3,017	1,371
Median Requests/Day	12,139	19,140	4,671	677
Unique Documents	12,235	16,834	10,116	7035
Total Storage Size (MB)	196	333	173	146
Mean Document Size (Bytes)	15,977	19,765	17,061	20,691
Coefficient of Variation	0.71	8.80	6.60	3.27
Total Bytes Transferred (MB)	1,750	919	272	158
Mean Transfer Size (Bytes)	15,997	9,393	12,883	16,499
Median Transfer Size (Bytes)	14,339	1,967	3,089	9,691
Coefficient of Variation	0.73	9.33	7.35	3.55

4.2.1 The Analysis of Client Workloads

The total requests and total bytes transferred by individual clients are characterized using a log-log complementary distribution of the sort used in [10]. We observed that the distribution of bytes transferred by individual clients is heavier-tailed ($\alpha = -1.24$) than the distribution of total requests generated by individual clients ($\alpha = -1.44$). The fact that both distributions are heavy-tailed indicates that a few clients are responsible for both the disk I/O and the CPU cycle bottlenecks. Therefore, understanding the reference patterns of these few clients may be particularly important to effective Web server resource management.

Table 2 gives statistical information of four selected client workloads that belong to the group of these few clients extracted from the aggregate server workload. These are identified as A, B, C, and D. Client A has the most requests per day and moves the most data. Client D references the fewest unique documents, but the documents tend to be larger. A weekend effect is observed in the workloads of clients B and C since their median requests per day is greater than their mean requests per day. More extensive information on these clients, including how they were selected, is given in [20].

4.2.2 The Reference Patterns of the Selected Clients

The reference patterns of the four selected client workloads is characterized in Figure 6 by plotting, on the y-axis, a sequence of numbers assigned to distinct documents found in each of the client workloads and, on the x-axis, the time since the start of the individual client requests. The terms "mostly working" and "mostly surfing" (as coined in [11]) are used in explaining each of the client reference patterns. A mostly working pattern implies that the client keeps re-referencing previously accessed documents most of the time, while a mostly surfing pattern implies that the client rarely keeps re-referencing previously accessed documents. Clients B and C display a mostly working pattern while clients A and D display a mostly surfing pattern. The dense reference pattern of client B indicates a strong working pattern while that of client C is weak. Since the rate of accessing new documents decreases as each day passes, temporal locality seems to be present in the reference patterns of clients B and C. It is speculated that client B is a firewall or relay, while client C is a time-sharing system. More specifically, client B is believed to access the server on behalf of human users with similar document reference patterns, while client C does the same for human users but with dissimilar document reference patterns. At any rate, their characteristics are quite different.

Similarly, the surfing reference pattern of client A reveals a crawling pattern that shows that the client accesses the same number of documents at the server each of the possible nine or ten times it visits the server during the period of observation. This is quite different from the surfing reference pattern of client D in which the first day appears to be spent accessing distinct documents and the subsequent days making scanty re-references to documents accessed during the first day. The crawling-surfing reference pattern of client A suggests that it is a non-human user, perhaps a search agent or crawler [2,3], while the pattern of client D, by contrast, suggests that it might be a caching proxy [4].

Clearly, these individual clients display quite different request characteristics. In the following, some of these characteristics are examined in more detail with consideration of the potential resource management implications of their differences.

[4] The caching proxy is empty on the setup day and the first document misses of 7,000 requests can be due to cold-start behaviour of an empty cache. The subsequent scanty reference pattern after the first day can be due to capacity misses resulting from a finite-sized cache at the proxy and coherent misses resulting from stale documents in the proxy cache.

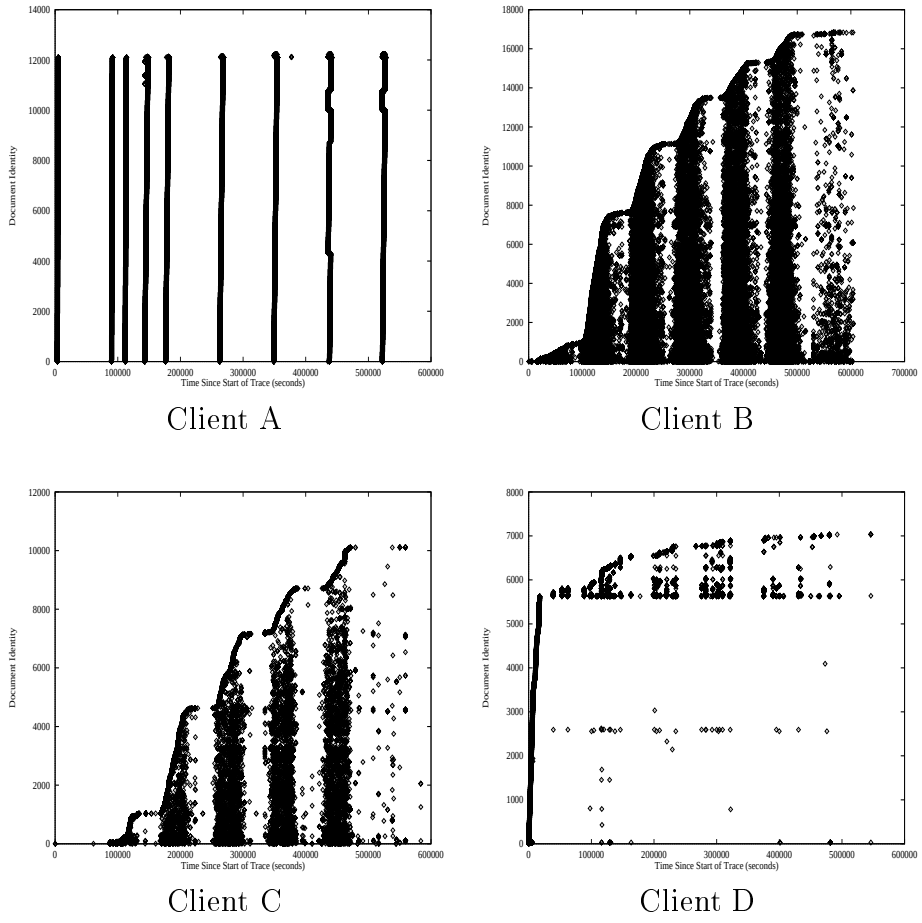


Fig. 6. Reference Patterns of Four Selected Clients

4.2.3 Concentration of Client Document References

The document referencing patterns of the selected clients are characterized in Figure 7 using Zipf's Law [23] as discussed in Section 4.1.2. We observed that client A has a uniform referencing behaviour, which alludes to a cyclic or crawling referencing pattern—the client visits all distinct documents at the server the first time and all subsequent referencing is to traverse this set of previously accessed documents [2,3]. The other clients B, C and D, however, reveal a non-uniform referencing pattern. The result is consistent with the hypothesis that client A is likely to be a non-human user such as a search agent or Web crawler, while the other clients are not. The implications of Figure 7 are that the reference pattern of client A is not amenable to caching, while the reference patterns of client B, C and D could be—because of the document concentration property reflected by their non-uniform referencing patterns.

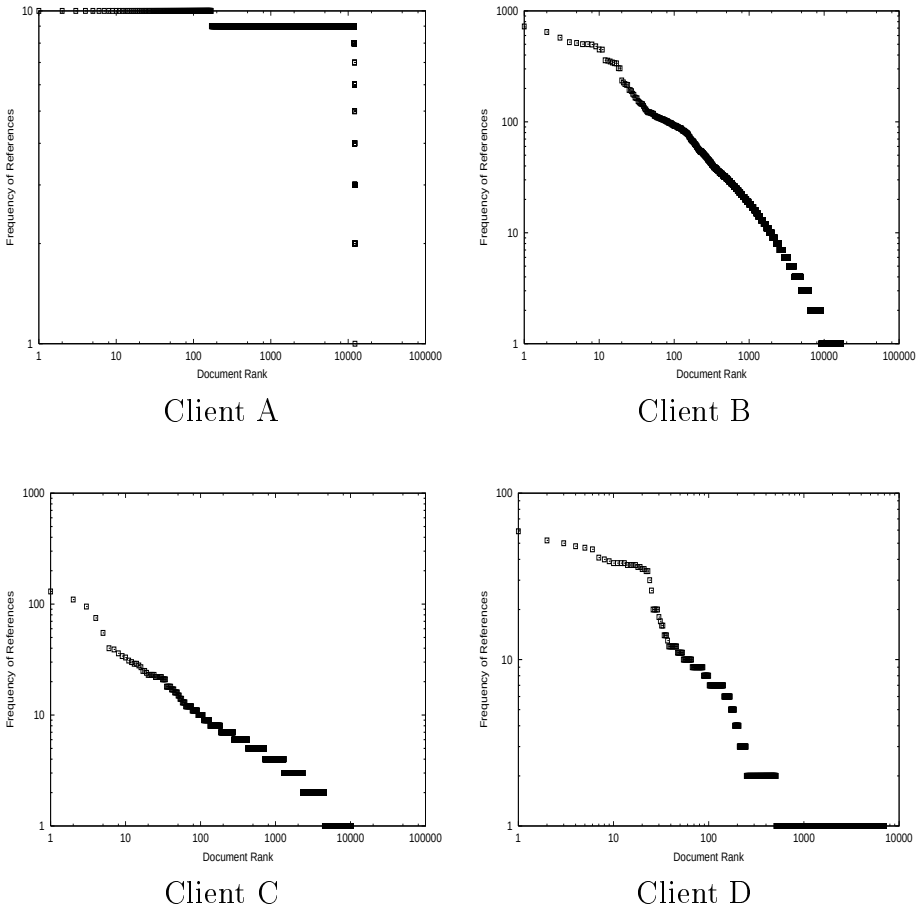


Fig. 7. Test for Zipf Distribution

4.2.4 Temporal Locality in the Selected Client Workloads

For caching the request streams of clients, temporal locality is an important consideration. The respective histograms of stack distances [5] (Figure 8) imply that client A has very poor temporal locality and clients B, C and D have weak temporal locality. For client A, most documents referenced occur at the same stack position, which is closer to the bottom of the stack. This further reflects the cyclic or crawling referencing pattern of a fixed set of documents by this client, leading to an inference of a non-human referencing pattern [2,3,5,6]. Caching the reference pattern of client A would require the cache to preserve some space ahead of time, or continuously free some space to accommodate the fixed set of documents accessed [2]. The reference patterns of the other three clients seem amenable to caching, despite the long tail observed in their stack distance distributions.

[5] A maximum stack length of 1000 yields a satisfactory measure of temporal locality for clients B, C and D. Client A requires an infinite stack due to its crawling document reference pattern.

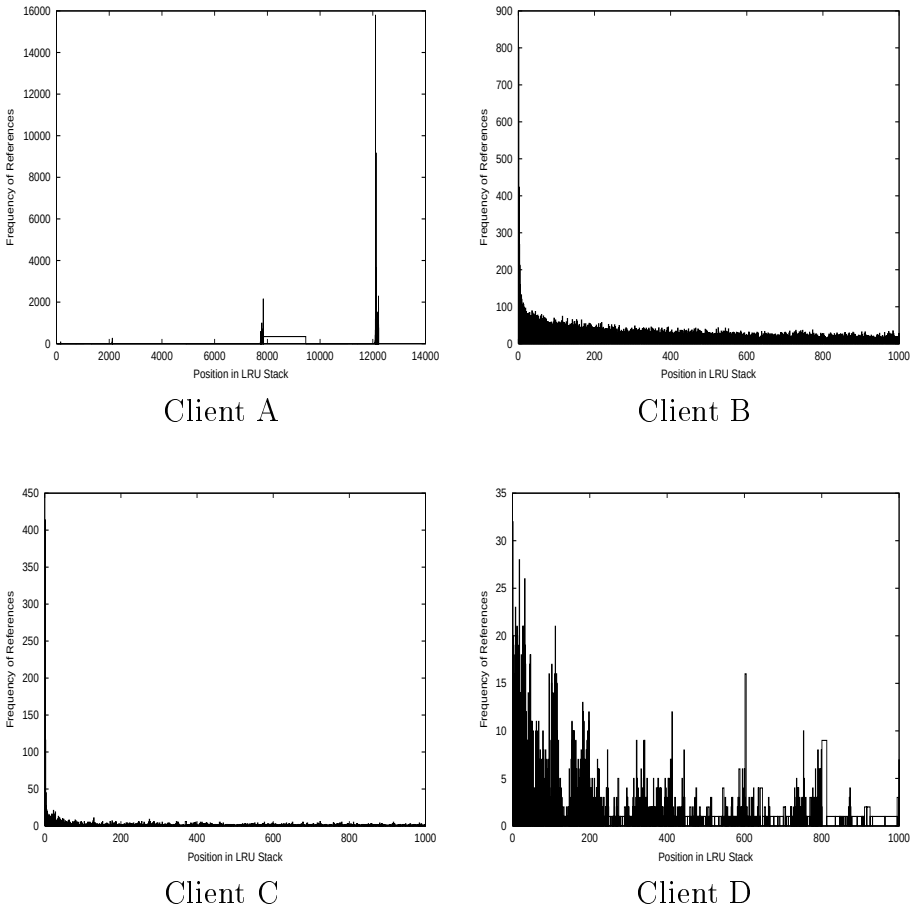


Fig. 8. Distribution of LRU Stack Depths

A relative measure of temporal locality was applied to the document reference pattern of the selected clients using the mean stack distance as reported in [7]. The results show that client B has the strongest temporal locality, followed by clients D, C, and A, respectively. In addition, the assumption of a "perfect cache", as discussed in [7,18], was used to verify the effectiveness of caching the reference patterns of these clients. The results indicate that only the reference patterns of client B and C are likely to be cacheable. The cacheability of client D's reference pattern cannot be ascertained because of the limited number of requests in its workload, but the reference pattern of client A is clearly not amenable to effective caching. Our findings are consistent with [2] which suggests that non-human clients such as crawlers have a referencing pattern that completely disrupts locality assumptions while human clients do not.

4.2.5 Inter-Arrival Times of Client-Requested Documents

The cumulative distributions of inter-request arrival times, time intervals between successive requests to documents from single clients, are shown in Fig-

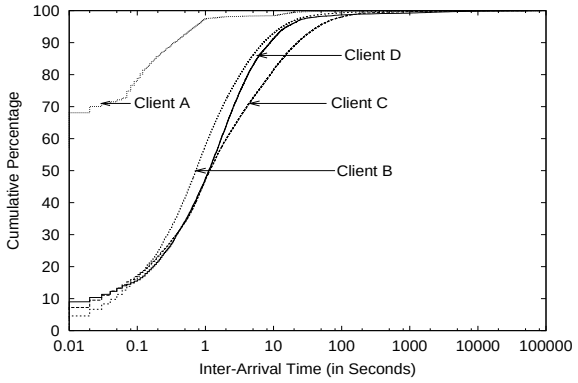


Fig. 9. Cumulative Frequency Distribution of Inter-Arrival Times for the Four Selected Clients

ure 9 for the four selected client workloads. Client A generates 95% of its requests to documents within one second, while the other clients (B, C and D) generate 80% and almost 100% of their requests within 10 and 100 seconds, respectively. This implies that a threshold of inter-request arrival time between 10 and 100 seconds would provide an effective timeout value for HTTP/1.1 persistent connections [6] for these clients. The impossibility that a human user can consistently generate requests within a second for a period of seven days adds weight to the conclusion that client A is a non-human client, while the others are likely to be human clients. In terms of usage of the HTTP/1.1 protocol, the reference pattern of client A is more closely aligned with the philosophy of persistent connections than other clients because of a smaller inter-request arrival time. This implies that non-human clients are more likely to benefit from persistent connections than are human clients.

4.2.6 Resource Management Implications

From the analysis of these four selected client workloads, two dramatically different reference patterns are apparent. These can be characterized as human and non-human. The reference pattern of human clients exhibits behaviour that is amenable to caching, but that of non-human clients does not. We speculate that these two referencing patterns represent the general referencing patterns of clients at the Web server, and see a need for resource management strategies that treat the request streams of human clients differently from those of non-human clients. Almeida *et al.* [2] suggested Web caching strategies that treat the request streams of non-human clients differently from that of human clients. We believe that caching the request streams of human clients and ignoring the request streams of non-human clients could be an effective

[6] This means that the number of TCP connections for the client-server communication on the Web has been reduced substantially by allowing a few clients to maintain the state of their open TCP connections at the server.

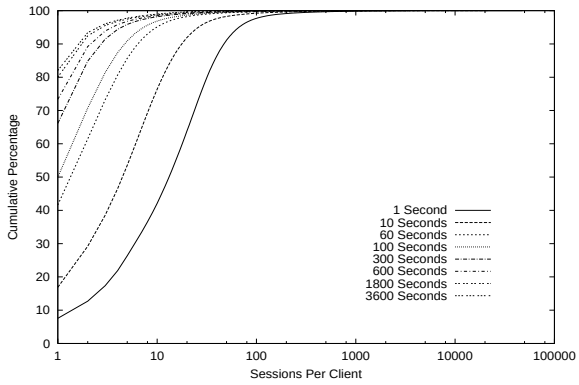


Fig. 10. Distribution of the Number of Sessions per Client

approach to Web server resource management. In general, we contend that resource management strategies at the Web server should be designed to use information about the characteristics of the clients being served.

4.3 Characteristics of Sessions within Individual Client Workloads

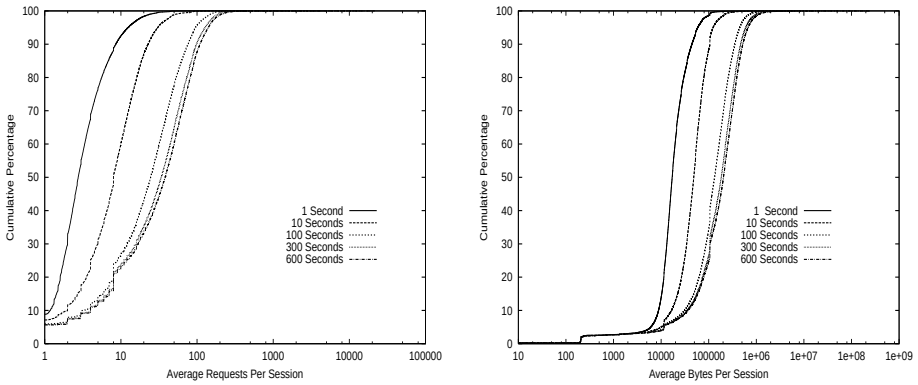
For this part of our investigation, we consider the requests comprising individual sessions within client workloads. Since a user's Web request behaviour can differ from one session to another, strategies for managing the reference pattern of clients and the HTTP/1.1 persistent connections at the Web server might be ineffective. Our objective is to characterize sessions within individual client workloads, focusing on the effectiveness of HTTP/1.1 persistent connections and the change in *document working set* [7] over sequences of sessions.

4.3.1 Web Session Analysis

We define a Web session (following the lead of [4]) as a stream of requests from a single client with an inter-request time less than a given threshold window time, or *timeout value*. If request r_{i+1} from a given client arrives at the server Δt seconds after request r_i from the same client, and $\Delta t \leq T$ (where T is the timeout value), then requests r_i and r_{i+1} are both considered to be part of the same client session. If $\Delta t > T$, then request r_i is the last request in the client session and indicates the end of a client session, while request r_{i+1} initiates the next client session.

In selecting a range of timeout values for characterzing sessions within client request streams, we examined the impact of fixed timeout values on the distribution of the number of sessions per client in the server workload (as shown

[7] The document working set is defined as the distinct documents referenced within a client session.



(a) (b)

Fig. 11. Analysis of Client Session Activity: (a) Average Requests per Session; (b) Average Bytes Transferred per Session

in Figure 10). Our observation suggests that as the timeout values increases, the number of sessions per client drops rapidly: for a timeout value of one second, about 10% of the clients have a single session; while about 50% of clients have at least 10 sessions. With a timeout value of 100 seconds, about 42% of the clients have a single session; while about 70% or more of the clients have a single session when the timeout value exceeds 100 seconds. The significant increase in the percentage of clients having a single session implies that most clients send few requests, and a few clients send most requests to the server. Since timeout values beyond 600 seconds have little or no impact on the distribution of the number of sessions per client, a maximum timeout value of 600 seconds is chosen for characterizing sessions within the request stream of clients.

4.3.2 Distribution of Requests and Bytes Transferred within Web Sessions

The effectivess of HTTP/1.1 persistent connections for the reference patterns of clients is evaluated by characterizing the number of requests and the volume of bytes transferred within sessions of individual client request streams in the server workload. Fixed timeout values between 1 second and 600 seconds were used. We observed (as shown in Figure 11) that a timeout value from 100 seconds and above is effective for HTTP/1.1 persistent connections because the distribution of either average requests or average bytes transferred within sessions of client request streams does not change significantly. For example, fewer than 10% of clients have an average of one request per session of their individual workloads for all timeout values considered, and 50% of clients transferred an average of 100-1000 kB per session for a timeout value of 100 seconds.

Since the statistical mean can lead to an unreliable measurement if the data is

skewed, the full distributions for total requests and volume of bytes transferred over sequences of sessions within individual clients at the server were analyzed (see [20] for details). We found a high variation in the total number of requests and the volume of bytes transferred across individual client sessions for timeout values of 1 and 10 seconds. Roughly 70% of the clients were observed to have a coefficient of variation that is close to 1.0. The variation decreases as the timeout value exceeds 100 seconds. Since a smaller timeout value is usually preferable owing to memory performance problems [4], a timeout value of 100 seconds offers good utilization of HTTP/1.1 persistent connections.

Applying the same method to the workloads of selected clients A, B, C and D, we observed that 95% of sessions within the request stream of client A account for at least 10 requests and more than 10 kB of documents transferred for all timeout values used. For a timeout value of 100 seconds, 60% and 85% of sessions within the request streams of clients B, C and D account for at least 10 requests and more than 10 kB of documents transferred from the server, respectively. Figure 12 shows the distributions for clients A and C (see [20] for detailed explanations). Overall result reveals that client A has the best utilization of HTTP/1.1 persistent connections for a fixed timeout value of 1 second compared to other clients with a fixed higher timeout value of 100 seconds. This implies that the reference pattern of non-human clients such as A can more effectively utilize HTTP/1.1 persistent connections at a fixed, but smaller timeout value than human clients such as B, C and D as alluded to in Section 4.2.5. We speculate that the timeout value of 100 seconds which yielded the best utilization of HTTP/1.1 persistent connections for all clients in this server workload is likely due to the dominant effect of client reference patterns which is human over that of non-human.

4.3.3 Phase Transition Behaviour within Web Sessions

The sequence of document working sets across sessions [8] of individual client workloads was characterized in an attempt to understand phases of client document referencing behaviour. The characterization shows the presence of phase transitions if there is a significant change in document reference patterns across sessions. Adapting Kienzle *et al.*'s [14] method for characterizing memory reference patterns to Web client reference patterns yields the cumulative frequency distributions of the percentage changes in document working sets across consecutive sessions shown in Figure 13. The percentage change in document working sets between consecutive sessions can be greater than 100% because the document working set of the current session can be larger than that of the previous session. The x-axis represents the percentage change

[8] The identification of sessions was based on the timeout values that provided the best utilization of persistent connections for these clients.

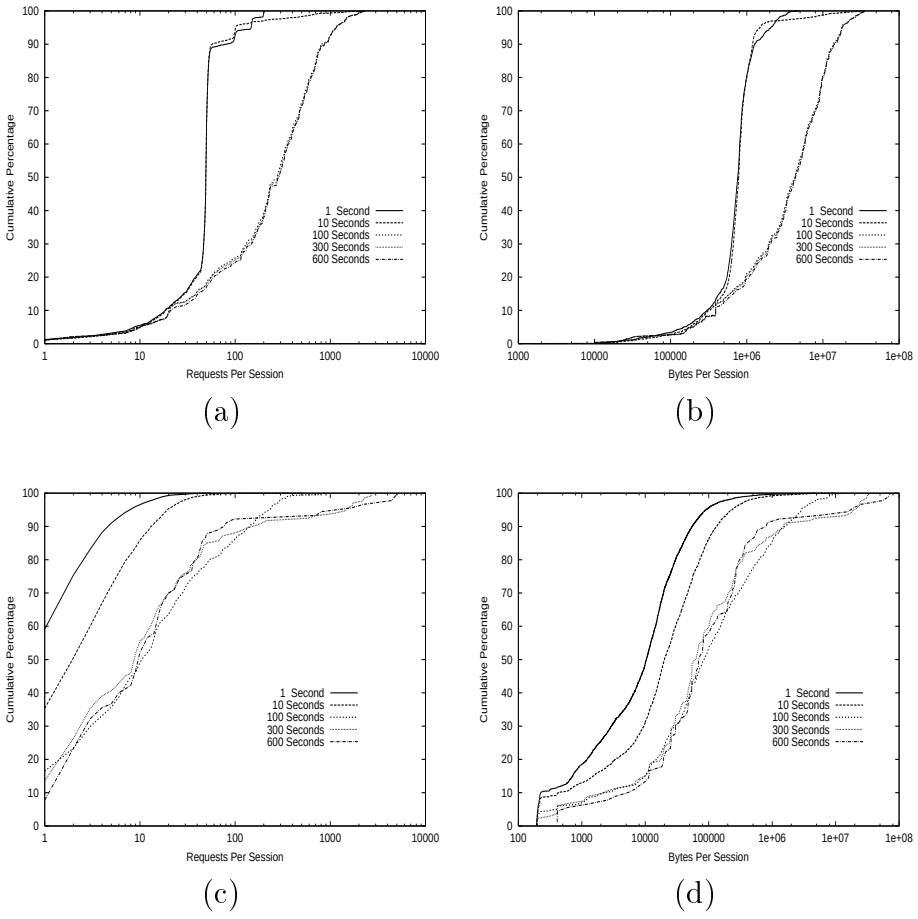


Fig. 12. (a-b) Distribution of Requests and Bytes Transferred within Sessions of the Request Stream of Client A; (c-d) Distribution of Requests and Bytes Transferred within Sessions of the Request Stream of Client C

in document working sets of consecutive sessions and the y-axis represents the percentage of consecutive sessions that exceeds a given percentage on the x-axis. Percentage changes that exceed 500% at the tail of the distributions are cut off for visualization purposes.

We observed that for client A, over 80% of consecutive sessions show a change in document working sets greater than 100%. For clients B, C and D, about 50% of their consecutive sessions have a change in document working sets exceeding 75%, 100% and 100%, respectively, after which a slowly declining pattern evolves. This pattern indicates a clear shift in the activity of clients B, C and D over their sessions. Since most consecutive sessions within the selected client workloads have a change in document working sets greater than 100%, phase transition behaviour is concluded to be present. This implies that the document reference patterns of sequences of sessions within individual client workloads may present problems for caching at the server.

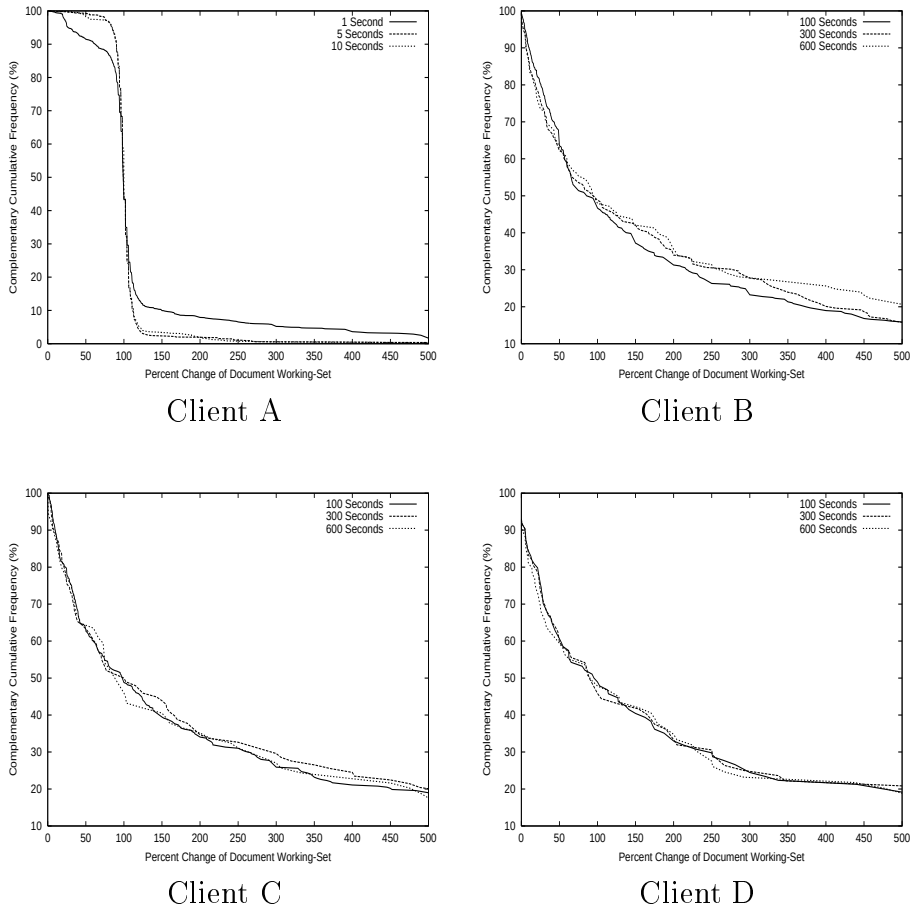


Fig. 13. Distributions of Percentage Changes Between Consecutive Document Working Sets of Client Sessions

4.3.4 Resource Management Implications

Arlitt [4] presents an extensive discussion of how the activity of clients using persistent connections [17,22] can affect the management of Web server resources such as memory. The explanation, however, hinges on how the Web server must reserve the memory resource in order to implement HTTP/1.1 persistent connections effectively. For this workload, the use of a fixed but short timeout value less than 100 seconds is found to under-utilize HTTP persistent connections. This in turn wastes the reserved memory resource at the Web server because there is no substantial reduction in the number of TCP connections. On the other hand, a fixed but high timeout value of 100 seconds is found to provide the best utilization of HTTP/1.1 persistent connections. But this fixed and higher timeout value can be effective for resource management only if the memory resource at the Web server is bottleneck-free. Since memory is not bottleneck-free, the current implementation of persistent connections in HTTP/1.1 is not likely to be effective at the server.

For effective resource management, our suggestion is that specific client information be made available to the Web server. This would assist in developing policies that use a fixed but short timeout value of 1 second to manage HTTP/1.1 persistent connections of non-human clients, while a dynamic but short timeout value (as suggested in [4,17]) can manage persistent connections of human clients. The presence of phase transition behaviour suggests that the implementation of caching strategies during HTTP/1.1 persistent connections is not likely to be effective at the Web server.

5 Conclusions

A hierarchical workload characterization was undertaken for a busy Web server. This adds to the growing body of knowledge on Web workload characteristics. The specific results presented can assist Web server designers in the provisioning of Web services as well as improving the effectiveness of new HTTP protocols.

Client differences can significantly impact performance. In particular, the identification of the reference patterns of human and non-human clients further helps in understanding the impact of these differences on performance. An important observation from this study is that information about clients visiting the Web server (whether human or non-human) can be beneficial to the effective management of server resources. We contend that Web servers should use information based on the reference patterns of isolated clients (such as human and non-human) to provide more effective resource management, and our future research is aimed at the development of such strategies.

5.0.5 Acknowledgements

An earlier version of this paper appeared in the proceedings of TOOLS'2002 Conference [19]. Greg Oster provided admirable technical support for this trace. Funding for this research was provided by the Natural Sciences and Engineering Research Council of Canada, through research grant OGP0003707, and Telecommunications Research Laboratories (TRLabs) in Saskatoon.

References

[1] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira, "Characterizing Reference Locality in the World Wide Web," in *Proceedings of the IEEE*

Conference on Parallel and Distributed Information Systems, Miami Beach, Florida, pp. 92-103, December 1996.

[2] V. Almeida, D. Menasce, R. Riedi, F. Pelegrinelli, R. Fonseca, and W. Meira, Jr., "Analyzing Web Robots and their Impact on Caching," in *Proceedings of the Sixth Workshop on Web Caching and Content Distribution*, Boston, Massachusetts, June 2001.

[3] V. Almeida, D. Menasce, R. Riedi, F. Pelegrinelli, R. Fonseca, and W. Meira, Jr., "Analyzing Robot Behaviour in E-Business Sites," in *Proceedings of the ACM SIGMETRICS Conference*, Cambridge, Massachusetts, pp. 338-339, June 2001.

[4] M. F. Arlitt, "Characterizing Web User Sessions," *Performance Evaluation Review*, Vol. 28, No. 2, pp. 50-56, September 2000.

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

[6] M. Arlitt and T. Jin, "A Workload Characterization Study of the 1998 World Cup Web Site," *IEEE Networks*, Vol. 14, No. 3, pp. 30-37, May/June 2000.

[7] P. Barford, A. Bestavros, A. Bradley, and M. Crovella, "Changes in Web Client Access Patterns: Characteristics and Caching Implications," *World Wide Web*, Vol. 2, No. 1, pp. 15-28, January 1999.

[8] H. Braun and K. C. Claffy, "Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server," *Computer Networks and ISDN Systems*, Vol. 28, pp. 37-51, 1996.

[9] L. D. Catledge, and J. E. Pitkow, "Characterizing Browsing Strategies in the World Wide Web," *Computer Networks and ISDN Systems*, Vol. 26, No. 6, pp. 1065-1073, 1995.

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

[11] C. R. Cunha and A. Bestavros, and M. E. Crovella, "Characteristic of World Wide Web Client-Based Traces," Technical Report TR-95-010, Department of Computer Science, Boston University, Boston, Massachussets, April 1995.

[12] J. H. Hine, C. E. Wills, A. Martel, and J. Sommers, "Combining Client Knowledge and Resource Dependencies for Improved World Wide Web Performance", in *Proceedings of the INET 1998 Conference*, Geneva, Switzerland, July 1998.

[13] S. Jin and A. Bestavros, "Sources and Characteristics of Web Temporal Locality," in *Proceedings of the Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecomminucation Systems*, San Francisco, California, August/September 2000.

[14] M. G. Kienzle, J. A. Garay, and W. H. Tetzlaff, "Analysis of Page-Reference Strings of an Interactive System," *IBM Journal of Research and Development*, Vol. 32, No. 4, pp. 523-535, July 1988.

[15] A. Mahanti, *Web Proxy Workload Characterization and Modeling*, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, September 1999.

[16] D. Menasce, V. Almeida, R. Riedi, F. Peligrinelli, R. Fonseca and W. Meira Jr., "In Search of Invariants for E-Business Workloads," in *Proceedings of the Second ACM Electronic Commerce Conference*, Minneapolis, Minnesota, October 2000.

[17] J. C. Mogul, "The Case for Persistent-Connection HTTP," in *Proceedings of the ACM SIGCOMM Conference*, Cambridge, Massachussets, pp. 299-313, August 1995.

[18] J. C. Mogul "Network Behavior of a Busy Web Server and Its Clients," WRL Research Report 95/4, Digital Western Research Laboratory, May 1995.

[19] A. Oke and R. Bunt, "Hierarchical workload characterization for a busy Web Server," *Proceedings of the 12th International Computer Performance Evaluation TOOLS Conference*, London, UK, April 2002.

[20] A. A. Oke, *Workload Characterization for Resource Management at Web Servers*, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, October 2000.

[21] J. Spirn, "Distance String Models for Program Behaviour," *IEEE Computer*, Vol. 9, No. 11, pp. 14-20, November 1976.

[22] K. Yap, "A Technical Overview of the New HTTP/1.1 Specification," in *Proceedings of the Third Australian World Wide Web Conference*, Australia, May 1997.

[23] G. K. Zipf, *Human Behaviour and the Principle of Least Effort*, Addison-Wesley, Cambridge, Massachusetts, 1949.