Temporal Locality and its Impact on Web Proxy Cache Performance [⋆]

Anirban Mahanti, Derek Eager, Carey Williamson

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

Abstract

This paper studies temporal locality characteristics present in document referencing behavior at Web proxies, and the impact of this temporal locality on document caching. First, *drift* measures are developed to characterize how the popularity profile of "hot" documents changes on a day-to-day basis. Experiments show that although there is considerable "hot set" drift, there is also a significant number of documents that have long-term popularity. Second, a measure of short-term temporal locality is developed that characterizes the relationship between recent past and near future document references. Using this measure, it is established that temporal locality arising out of the correlations between document references in the recent past and near future does exist for popular documents.

Another objective of this paper is to determine whether or not Web document references at proxy caches can be modeled as independent and identically distributed random events. Trace-driven simulations using empirical and synthetic traces (with varying degrees of temporal locality) show that temporal locality is an important factor in cache performance. The caching simulation results also show that temporal locality arising out of short-term correlations between references is important only for small caches. For large caches, a synthetic workload generated by applying the Independent Reference Model on a day-to-day basis gives performance very similar to that obtained for empirical traces.

1 Introduction

The World-Wide Web (WWW, or "Web") has experienced phenomenal growth in recent years. This growth has contributed significantly to the network traffic

[⋆] Appears in Performance Evaluation, Special Issue on Internet Performance Modelling, September 2000 on the Internet, and motivated much research into improving the performance and scalability of the Web.

A popular and effective technique for improving Web performance is caching. Caching is effective because a small fraction of the total documents accessed on the Web often account for a large fraction of document references. By caching popular documents closer to the requesting clients, network traffic can be greatly reduced.

In recent years, Web proxies have been deployed to reduce network traffic and provide better response time for Web accesses. A Web proxy consists of application level software that accepts document retrieval requests from a set of clients, forwards these requests to appropriate servers if the requested documents are not already present in the proxy's cache, and sends documents back to the clients [12]. Proxies were originally designed to allow network administrators to be able to control access to the Internet from within an Intranet [4]. It was recognized, however, that proxies may also serve as repositories for frequently requested documents. Caching documents at the proxy can save network bandwidth and reduce network latency for document accesses [13, 15].

Experience with traditional computer system workloads has shown the importance of *temporal locality* on computer systems design [21]. Intuitively, temporal locality refers to the characteristic of the reference stream, in which there is an increased tendency to reference in the near future elements referenced in the recent past. In the Web context, temporal locality can be exploited in the design of better caching and prefetching systems. Although there has been considerable interest in understanding Web document reference characteristics [1, 2, 5, 6], and Web caching [14, 20, 25], the presence and importance of temporal locality is still the subject of some debate [1, 7, 9].

In this paper, we investigate temporal locality characteristics over both long and short time intervals. From our results, it is evident that temporal locality does exist in Web proxy workloads. This observation raises two important questions: (a) "Does temporal locality really matter in Web proxy cache performance?"; and (b) "What is the source of the observed temporal locality?". If temporal locality is unimportant for Web proxy cache performance, then generating synthetic workloads would be rather straightforward. Otherwise, synthetic workload generation needs to incorporate temporal locality in the synthetic traces. Furthermore, understanding the source of temporal locality can provide useful insight into the present state of caching in the Web.

To address these questions, a synthetic trace generator was developed using the Independent Reference Model (IRM) [10]. Caching simulations with the empirical and synthetic traces suggest that temporal locality is indeed important for analyzing quantitative aspects of Web cache behavior. However, we also observe that (for large caches) temporal locality arising out of short-term correlation between document references has little impact on cache performance. Finally, we conjecture that most of the observed short-term temporal locality arises due to inconsistencies in caching documents at the browsers and proxies.

The rest of the paper is organized as follows. The next section provides a brief description of related work. Section 3 describes the data collection sites and the data reduction process. Section 4 presents the temporal locality analysis of the Web proxy workloads. Section 5 studies the impact of temporal locality on Web cache performance. Causes of the observed short-term temporal locality in the document reference streams at Web proxies are considered in Section 6. The paper concludes in Section 7.

2 Related Work

The operating systems research community has devoted much time and effort to studying the locality characteristics of memory and file reference patterns. Denning and Schwartz established the fundamental properties of locality as applied to memory reference patterns [10]. Similar characteristics have been observed for file reference patterns [18, 21]. Later, the advent of distributed systems consisting of workstations and shared file servers resulted in much research on locality characteristics and their impact on caching at client [3] and file server caches [26].

Many recent studies have focussed on the characteristics of Web traffic at clients [5], proxies [16, 28], and servers [1, 2, 6]. Almeida *et al.* [1] used the LRUSM model to measure temporal locality in Web server access logs. Cao *et al.* [9] analyzed document inter-reference times to establish the presence of temporal locality in Web proxy access logs. Others have used trace-driven caching simulations to study the influence of workload characteristics (including locality) on Web document caching at proxies [9, 20]. A more recent study analyzed how workload characteristics change across different levels of a proxy caching hierarchy [17].

Our work builds upon these previous research efforts. The focus is on developing flexible and easy to interpret measures for characterizing temporal locality in the Web context, understanding the impact of this temporal locality on document caching, and determining the source of the observed temporal locality.

3 Data Collection and Reduction

Data Collection Sites: The access logs for this study were obtained from two Web proxy servers: the proxy server at the University of Saskatchewan, and the National Laboratory for Applied Network Research (NLANR) proxy [19] at the University of Illinois, Urbana-Champaign. These access logs provide information on proxy servers with quite different workloads. The clients of the University of Saskatchewan proxy server are mostly individual users. The NLANR proxy cache is one of several top-level nodes in the NLANR Web caching hierarchy; it receives requests from sibling caches at the top level, as well as from lower-level caches for which it is a parent. Most of the clients of the NLANR cache at Urbana-Champaign are institutional-level caches located in the United States.

The access logs of individual days were concatenated to obtain longer data sets for each site. A 21-day trace[1] of references to the University of Saskatchewan proxy server was created by concatenating access logs from March 1 to March 21, 1999. Similarly, a 21-day NLANR trace spanning August 30 to September 19, 1999 was created. The University of Saskatchewan proxy server recorded a total of 16,406,920 requests in 21 days of activity. The NLANR proxy recorded 28,956,442 requests in the 21 days of activity. These data sets will be referred to as USask and NLANR in the rest of this paper.

Data Reduction: Each access log records one line of information per request processed at the proxy server. An entry in the access log records the URL of the document being requested, the date and time of the request, the name (or the IP address) of the client making the request, the number of bytes returned to the requesting client, and additional information that describes how the client's request was treated at the proxy.

The traces were pruned to contain only the information useful for our study. For example, a preliminary analysis revealed that the USask proxy was configured to record inter-cache queries made using the Internet Cache Protocol (ICP) [24]. Since ICP queries do not result in document transfers, these entries were removed from the traces. The HTTP reply codes in the access logs were employed for further data reduction. The HTTP reply codes describe how a client's request was serviced. In this study, we consider all requests that would result in the document being accessed from the origin server in the absence of intermediate proxies. Therefore, only requests with the 200 (OK) and 206 (Partial Content) status-codes are considered.

The next step in the data reduction process was to discard requests for dy-

[1] Larger data sets were used in [16]. However, the results presented here are for a more recent trace, and are consistent with the observations made in [16].

namic documents, since these documents are typically not cached at proxies. We assumed that all documents with a "cgi-bin" or "?" in the URL string represent dynamic content. These documents account for less than 2% of the HTTP requests recorded in the logs.

The final step in our data reduction process was to identify uncacheable (static) documents [8, 27, 28]. Since the access logs do not provide the required information (e.g., last modified dates, set-cookie headers, no-cache pragmas), we developed an *ad hoc* strategy to filter the uncacheable documents. We replayed the (partly) reduced access logs through our proxy cache simulator (described later) under different cache sizes using an LRU replacement policy. A *false hit* is said to occur when a reference to a particular document results in a hit at the simulated proxy, but is recorded as a miss in the access logs. The number of false hits, expressed as a fraction of the total number of requests, ranged from 3.0% (256 MB cache) to 6.6% (64 GB cache) for the USask data set, and from 5.0% (256 MB cache) to 7.4% (64 GB cache) for the NLANR data set. Based on the assumption that, for simulated cache sizes at least as large as the actual proxy cache size, a false hit indicates an uncacheable document, we removed from the access logs those documents for which a false hit was recorded when using an 8 GB cache for the USask data set, and a 16 GB cache for NLANR [2].

Table 1 summarizes the characteristics of the two reduced traces used for the analysis and simulations in this paper.

4 Temporal Locality in Web Proxy Workloads

Intuitively, temporal locality refers to the property that referencing behavior in the recent past is a good predictor of the referencing behavior to be seen in the near future. While temporal locality is traditionally used to describe the referencing behavior in an aggregate reference stream, it can also be considered on a per-item basis. In the latter context, temporal locality refers to the property that the probability of referencing a particular item decreases with an increasing time of last reference to that item. It is worth noting that the notion of temporal locality is orthogonal to the *concentration*[3] behavior analyzed in [2, 16] in the sense that the presence of concentration need

[2] At the time our access logs were captured, the USask proxy was using a 5 GB disk cache, and the NLANR proxies typically allocated 10 GB of the disk for caching. Since the simulated proxy caches are larger than the actual cache sizes, our estimate of uncacheable documents is conservative.

[3] Concentration of references implies that a small fraction of the total documents account for a large fraction of the total references.

Table 1
Summary Characteristics of the Reduced Traces

Item	USask	NLANR
Trace Duration	21 days	21 days
Start Date	Mar 1, 1999	Aug 30, 1999
End Date	Mar 21, 1999	Sept 19, 1999
Total Requests	10,371,995	17,638,168
Avg Requests/Day	493,904	839,912
Total Bytes Transferred (GB)	83.49	222.45
Avg Bytes/Day (GB)	3.98	10.59
Distinct Documents	3,166,848	5,517,024
Distinct Documents/Total Reqs. (%)	30.53	31.28
One-Timer Documents	2,228,595	3,834,753
One-Timers/Requests (%)	21.47	21.74
One-Timers/Distinct Documents (%)	70.37	69.51
Total One-Timer Bytes (GB)	26.70	74.20

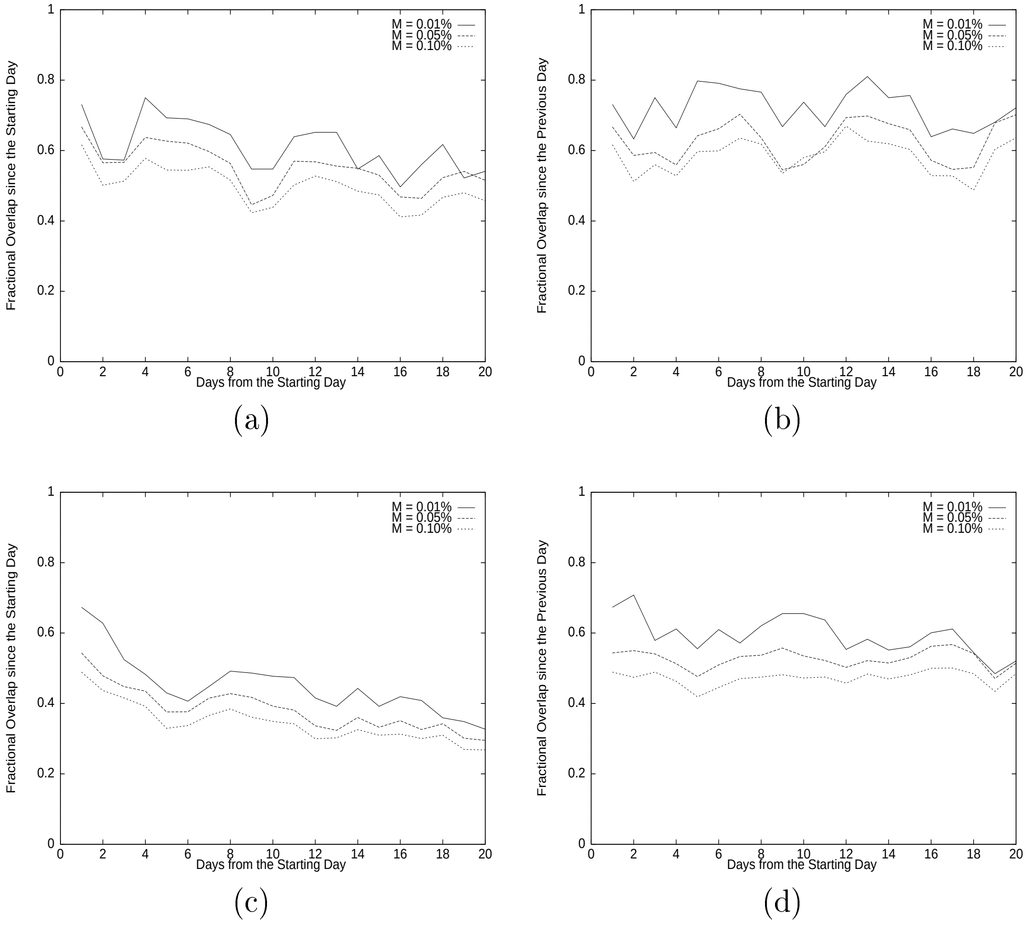
not necessarily imply the presence of temporal locality (i.e., the concentration metric does not consider the correlation between a reference to a document and the time since it was last accessed). The following analysis focuses on the presence of "long-term" and "short-term" temporal locality in the workloads.

4.1 "Hot Set" Drift Analysis

A simple analysis was performed to understand how the set of most popular documents (referred to as the "hot set") changes with time for each Web proxy. The objective of this analysis is to determine whether or not "long-term" temporal locality is present in Web proxy references.

Two different measures called *absolute* and *relative* drift are developed. Considering a "hot set" of size "M" (i.e., the set of M most frequently referenced documents), *absolute drift* measures the fractional overlap between the "hot set" on day i of the trace and the "hot set" on the initial starting day (day 0). Similarly, *relative drift* measures the fractional overlap between the "hot set" on day i of the trace and the "hot set" on the previous day (i.e., on a day-to-day basis). To understand the impact of the size of the "hot set" on the drift measures, three values for M are considered: 0.01%, 0.05%, and 0.10% of the total unique documents in each trace.

The results of the “hot set” drift analysis are presented in Figure 1. Ignoring the obvious non-stationarities due to “week-end” effects, the relative drift in the “hot set” appears to be fairly constant for a particular value of M. The gradual absolute drift of the “hot set” reflects the enduring popularity of some documents. Also, note that the fractional overlap of documents decreases as the size of the “hot set” increases, suggesting the presence of a small but significant number of documents with enduring popularity.



(a)

(b)

(c)

(d)

Fig. 1. “Hot Set” Drift Results: (a) Absolute Drift for the USask data set; (b) Relative Drift for the USask data set; (c) Absolute Drift for the NLANR data set; (d) Relative Drift for the NLANR data set

It can also be observed that the “hot set” drift is more rapid for the NLANR data set compared to the USask data set. A possible explanation for this behavior is that references to more persistent “hot” documents tend to get captured at the lower-level caches.

4.2 Short-Term Measure of Temporal Locality

The *Least Recently Used Stack Model* (LRUSM) [22] has been widely used [1, 2, 26] to measure temporal locality. The LRUSM is a stack-based ordering of referenced objects, according to their recency of reference (i.e., most recently referenced item on top (position 1), and the least recently referenced item on the bottom). For each reference in the request stream, the stack is searched from top to bottom until the requested object is found, or the bottom of the stack is reached. If found (i.e., a hit), the object is moved from its present position in the stack (say, d) to the top of the stack, pushing the other $d-1$ items that used to be above it down one position. For an item that is not found in the stack (i.e., a miss), it is simply added to the top of the stack, pushing all other stack items down one position. Of interest is the probability of referencing a particular position in the stack (and not a particular document).

The main drawback of the LRUSM model is its inability to distinguish "hot set" effects from short-term temporal locality. That is, a "hot" document may cause many references to near the top of the stack, even if there is no correlation between the probability of referencing a particular document and the time since the document was last referenced.

To address this drawback, we develop a quantitative measure of temporal locality that separates "hot set" effects from temporal locality . This measure is defined for a particular document D_i at stack depth (or "buckets" of stack depths) j as follows:

$$T_{ij} = \frac{fraction\ of\ references\ to\ D_i\ made\ when\ D_i\ is\ at\ stack\ depth\ j}{fraction\ of\ total\ references\ made\ when\ D_i\ is\ at\ stack\ depth\ j}$$

Our measure is an indicator of whether the probability that a reference is to a particular document is independent of the probability of finding that document at a particular stack position (i.e., independent of the recency of past references to the document under consideration). Values of T substantially greater than 1 at low stack depths suggest the presence of short-term temporal locality in the referencing characteristics; lower values of T do not.

The short-term temporal locality measure T was computed for the 25 most popular documents for the first day in each of the traces. The top 50 positions in the LRU stack were considered and grouped into buckets of size 5 (i.e., stack depth 1-5 is bucket 1, stack depth 6-10 is bucket 2, and so on).

Table 2 shows the values of T for the ten most popular documents in the data sets (results for the next fifteen most popular documents are similar). The results indicate that there is considerable temporal locality in the referencing

Table 2
Short-Term Temporal Locality Measure for the Ten Most Popular Documents

USask										
Document	Bucket 1	Bucket 2	Bucket 3	Bucket 4	Bucket 5	Bucket 6	Bucket 7	Bucket 8	Bucket 9	Bucket 10
1	12.80	8.95	9.68	8.65	6.86	6.50	6.14	6.55	5.32	4.33
2	73.00	81.86	55.41	47.86	36.69	54.19	10.32	0.00	13.38	0.00
3	10.80	6.09	5.59	3.86	3.36	3.73	2.27	3.76	2.96	3.10
4	14.76	7.54	5.86	4.18	2.56	2.62	1.15	1.25	0.88	0.59
5	0.60	0.80	1.51	1.43	0.82	1.24	1.56	0.94	0.32	0.84
6	1.28	0.99	0.85	1.42	0.72	1.01	1.30	1.59	0.73	1.03
7	1.70	0.71	1.85	1.16	0.72	1.31	1.03	0.89	1.77	0.74
8	0.86	0.71	1.56	1.58	1.01	1.45	1.02	1.02	1.32	1.20
9	1.69	1.55	0.99	0.86	0.58	1.45	1.47	1.34	1.19	1.36
10	1.01	1.01	0.72	1.31	1.46	1.33	1.03	1.04	1.35	0.76

NLANR										
Document	Bucket 1	Bucket 2	Bucket 3	Bucket 4	Bucket 5	Bucket 6	Bucket 7	Bucket 8	Bucket 9	Bucket 10
1	2.44	2.16	2.34	1.42	1.75	0.64	0.65	1.63	1.48	1.82
2	1.71	1.53	0.77	0.77	1.36	2.16	1.59	2.00	2.64	3.29
3	0.79	1.98	1.00	1.01	1.62	2.66	3.10	2.32	2.13	2.16
4	1.11	1.11	0.75	1.50	0.00	0.00	0.37	0.75	2.27	2.67
5	0.76	1.53	0.38	0.77	0.77	0.39	0.39	1.16	1.56	1.96
6	6.66	2.38	4.43	6.56	3.36	3.83	3.44	4.40	3.56	3.15
7	2.88	0.83	2.48	0.83	1.68	1.69	2.12	1.71	0.43	3.46
8	0.00	0.00	1.60	2.41	0.00	1.61	1.62	0.81	0.81	0.81
9	0.00	0.80	0.80	0.00	0.00	0.00	0.80	2.43	0.81	0.00
10	1.63	5.76	1.66	1.67	1.68	0.84	1.68	0.85	2.55	0.85

behavior for some, but not all, of the most popular documents (e.g., documents 1-4 in the USask data set, and document 6 in the NLANR data set).

The referencing pattern of a document causes its T measure to increase or decrease at particular stack positions. Notice that the T measure for many documents (e.g., document 1 in the USask data set) generally appears to decrease as one moves deeper into the stack. This indicates the presence of short-term temporal locality in which the probability of re-reference is a decreasing function of the time since last reference. On the other hand, the T measure for some documents (e.g., document 3 in the NLANR data set) increases as one moves deeper into the stack. Such temporal locality characteristics are observed when a document does not stay for a long duration at lower stack positions without being referenced again. This type of referencing behavior can be caused by documents that have associated pages (e.g., embedded images). For example, some HTML pages have multiple references to a particular image file. When such pages are requested by the browsers, repeated references to the same image file are seen, separated by references to other embedded objects.

5 Impact of Temporal Locality on Web Proxy Cache Performance

5.1 *Synthetic Traces*

To determine the impact of temporal locality on the cache performance achieved by different replacement policies, trace-driven caching simulations using empirical and synthetic traces were carried out. Each empirical trace is a time-ordered file identifying all documents referenced along with the associated transfer size[4], for a particular duration of activity seen at the Web proxy. This section outlines the basic model for the synthetic traces, describes the synthetic workload generation process, and the validation of the synthetic traces.

5.1.1 *The Independent Reference Model*

The synthetic traces are based on the *Independent Reference Model* (IRM) [10]. According to this model, the document reference stream $D_1, D_2, ..., D_t, ...$ is considered to be a sequence of independent random variables with stationary probabilities: $P\left[D_t = i\right] = \lambda_i, 1 \leq i \leq n, t > 0$, where n is the number of documents, and λ_i is the (stationary) probability of referencing document i.

With the IRM model, the distribution of the inter-reference distance is geometric. That is, the probability $d_i(k)$ of an inter-reference distance k for a document i is $\lambda_i(1-\lambda_i)^{k-1}$ [10]. The overall inter-reference distance probability function $d(k)$ (i.e., the probability that another request for a document is separated by $k-1$ requests for different documents) is $d\left(k\right) = \sum_{i=1}^{n} \lambda_i d_i\left(k\right)$.

5.1.2 *Synthetic Workloads*

A synthetic trace generator was developed that takes an empirical trace file as input, and generates the following two synthetic traces:

- `Synthetic1`: This trace preserves the referencing probabilities of the documents in the empirical trace, but filters out any long-term and short-term temporal locality. This is achieved by generating the synthetic trace in a random order, according to fixed static probabilities based on the frequencies of reference in the empirical trace file.
- `Synthetic2`: In addition to preserving the overall referencing probabilities of the documents, this trace also preserves any long-term temporal locality

[4] Since individual document sizes are not recorded in the access logs, the transfer sizes are used as document sizes in the simulation.

observed in the original trace, by generating references on a day-to-day basis (within a day, in a random order) using the empirical referencing probabilities measured for each day.

When generating the synthetic traces, a reference is defined to be a "two-tuple" consisting of a document and its associated transfer size. This definition of a *unique* reference assumes significance in the Web context because a considerable fraction of the documents experience modifications [16]. The synthetic trace generator *does not* change the order in which the transfer size changes (and thus the presumed document modifications) occur in the empirical traces. In this manner, the pattern of assumed document modifications is preserved.

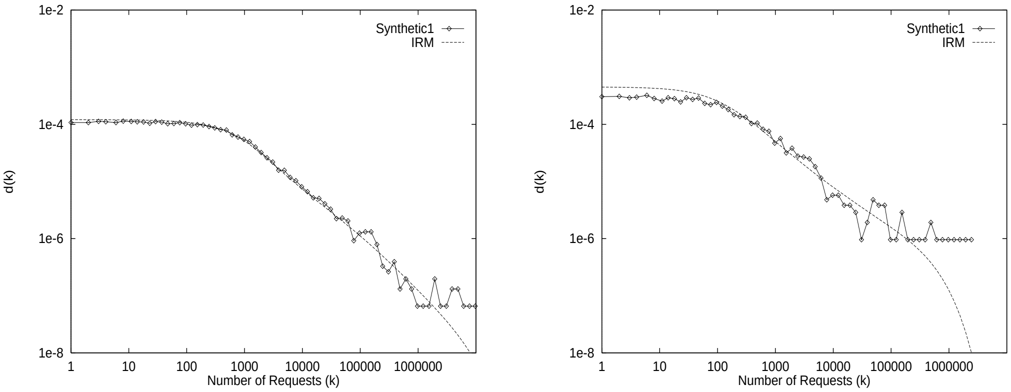
Almeida *et al.* [1] generated a synthetic workload similar to the `Synthetic1` trace to understand the applicability of the IRM model in the Web server environment. However, the approach used here is different in two respects. First, document modifications are explicitly modeled, since these can substantially affect the achieved hit ratios. That is, the synthetic workload is *not* generated by applying random permutations to the original trace, as in [1]. Second, caches of fixed size (in bytes) are simulated, instead of caches with storage capacity of a fixed number of documents (as in [1]).

5.1.3 Synthetic Trace Validation

The validity of the synthetic traces was established by comparing the document inter-reference distance probability distributions observed in the synthetic traces with those predicted by the IRM model. One-timers (i.e., documents with no repeated references) were not considered in this comparison.

Figure 2 shows that the document inter-reference distance probabilities for the `Synthetic1` traces closely follow the IRM model. The document inter-reference density function for the IRM model assumes an unending sequence of random events with stationary probabilities. However, real traces are of finite length, therefore, a limit is placed on the maximum inter-reference distance. A direct artifact of finite trace duration is a "flattening" of the distribution for large inter-reference distances in the figures. Since the IRM model is applied on a day-to-day basis when generating the `Synthetic2` traces, the document inter-reference distribution of a particular day in the synthetic trace is compared with that predicted by the IRM model for the same day. This comparison yields results similar to those shown in Figure 2 [16].

The short-term temporal locality measure T was computed for the ten most popular documents in the `Synthetic1` and `Synthetic2` traces [16]. These values were found to be (statistically) close to 1.0, further confirming the absence of short-term temporal locality in these versions of the traces.



(a) (b)

Fig. 2. Document Inter-reference Distance Probability Density Function, `Synthetic1` Trace versus IRM Model: (a) USask; (b) NLANR

5.2 Experimental Design

5.2.1 Simulation Model

The simulation model used in this study is similar to the ones used by other researchers to evaluate the performance of Web proxy cache replacement policies. The input to the simulated system is a set of time-ordered document retrieval requests. Upon receiving a request, the proxy searches its cache for the requested document. A *cache hit* occurs if the requested document is served from the cache; otherwise, a *cache miss* occurs. A cache miss can occur for the following reasons: (a) the requested document is not found in the cache; or (b) the cache has a *stale* copy of the requested document in its cache (i.e., the document has been modified since it was last cached). In the simulations, a cached copy of a document is defined as stale, if the document size associated with the request (in the trace file) is different from the size of the document in the cache. On a cache miss due to document modification, the stale copy of the document is purged from the cache, and a copy of the updated document is added to the cache. In order to create sufficient space to add a new document to the cache, the removal of zero or more documents from the cache may be required. Documents larger than the specified cache size are never cached.

5.2.2 Experimental Factors and Parameters

The number of factors (system and workload parameters), and the levels associated with each factor, have a great impact on the number of experiments that must be performed. Two system parameters are considered in this study: cache size and the cache replacement policy.

Cache Size: A total of 12 levels are considered in this study, ranging from 1 MB (megabyte) to 256 GB (gigabytes). The largest cache size of 256 GB is large enough to hold the entire document set for any of the traces considered. Therefore, results for this size indicate performance for a cache of unbounded size.

Cache Replacement Policy: In recent years, many proxy cache replacement policies have been proposed [9, 20, 25]. To make the number of experiments manageable, only three different replacement policies are considered: LRU, LFU-Aging, and GD-Size(1) [9]. The rationale behind selecting these caching algorithms is that they reflect a broad range of cache replacement policies, namely recency-based, frequency-based, and size-based, respectively. Readers interested in the implementation details of these algorithms are referred to [16].

Workload Parameters: The workload parameters considered in this study are the set of client requests and the length of the "warm-up" period. As outlined in Section 5.1.2, access logs from the Web proxy sites are used to create the empirical and the synthetic trace files. In this study, the effect of varying the client population making requests to the proxy (as in [11]) is not considered. Instead, the entire client set is used for the simulation experiments.

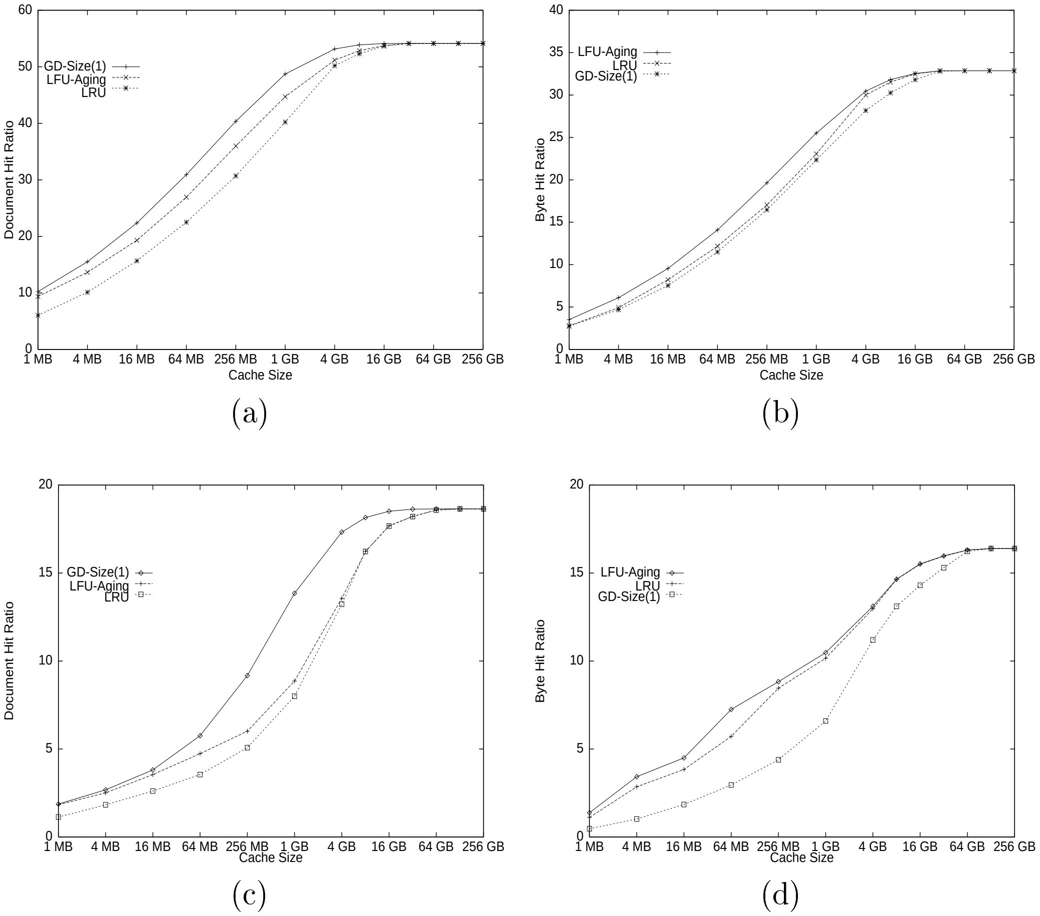
Most cache simulation studies focus on the steady-state behavior of the system. The idea is to neglect the cache misses due to an initially empty cache, as the replacement policy has no role to play in this case. Since the objective here, however, is to compare cache performance with the synthetic traces and the empirical trace as input, statistics are collected even when the cache is "warming up". Thus, in particular, an infinite cache simulation will achieve the same hit ratios for both the synthetic and the empirical traces.

5.2.3 Validation of the Simulator

An important step in any simulation study is to validate the simulator. To ensure that the simulator performed properly, test runs were carried out using two short traces (100 - 150 requests). One was a hand-made trace while the other was part of the empirical trace. The results of the simulator were then verified by hand. Also, obtaining the same hit ratios for infinite cache simulations with different replacement policies enhanced confidence in the simulator. Furthermore, the hit ratios reported here are similar to those obtained by other researchers [9, 11, 14].

5.3 Simulation Results

The cache performance with the three replacement policies considered in this study for the USask and NLANR traces is shown in Figure 3. Figure 3(a) shows that the maximum possible document hit ratio for the USask data set is about 54%; the remaining 46% of the requests are for documents requested for the first time, or for documents that have been updated. Since compulsory misses account for 31% of the total requests (see Table 1), the percentage of misses owing to document updates is 15%. Figure 3(b) shows that the maximum possible byte hit ratio for the USask data set is about 33%. The higher-level NLANR proxy achieved much lower cache hit ratios compared to the USask proxy, probably because of caching of popular documents at lower-level (e.g., institutional) caches. The maximum achievable document hit ratio and byte hit ratio for the NLANR proxy are about 19% and 16%.



(a)

(b)

(c)

(d)

Fig. 3. Cache Performance Results with the Three Replacement Policies: (a) Document Hit Ratio (USask); (b) Byte Hit Ratio (USask); (c) Document Hit Ratio (NLANR); (d) Byte Hit Ratio (NLANR)

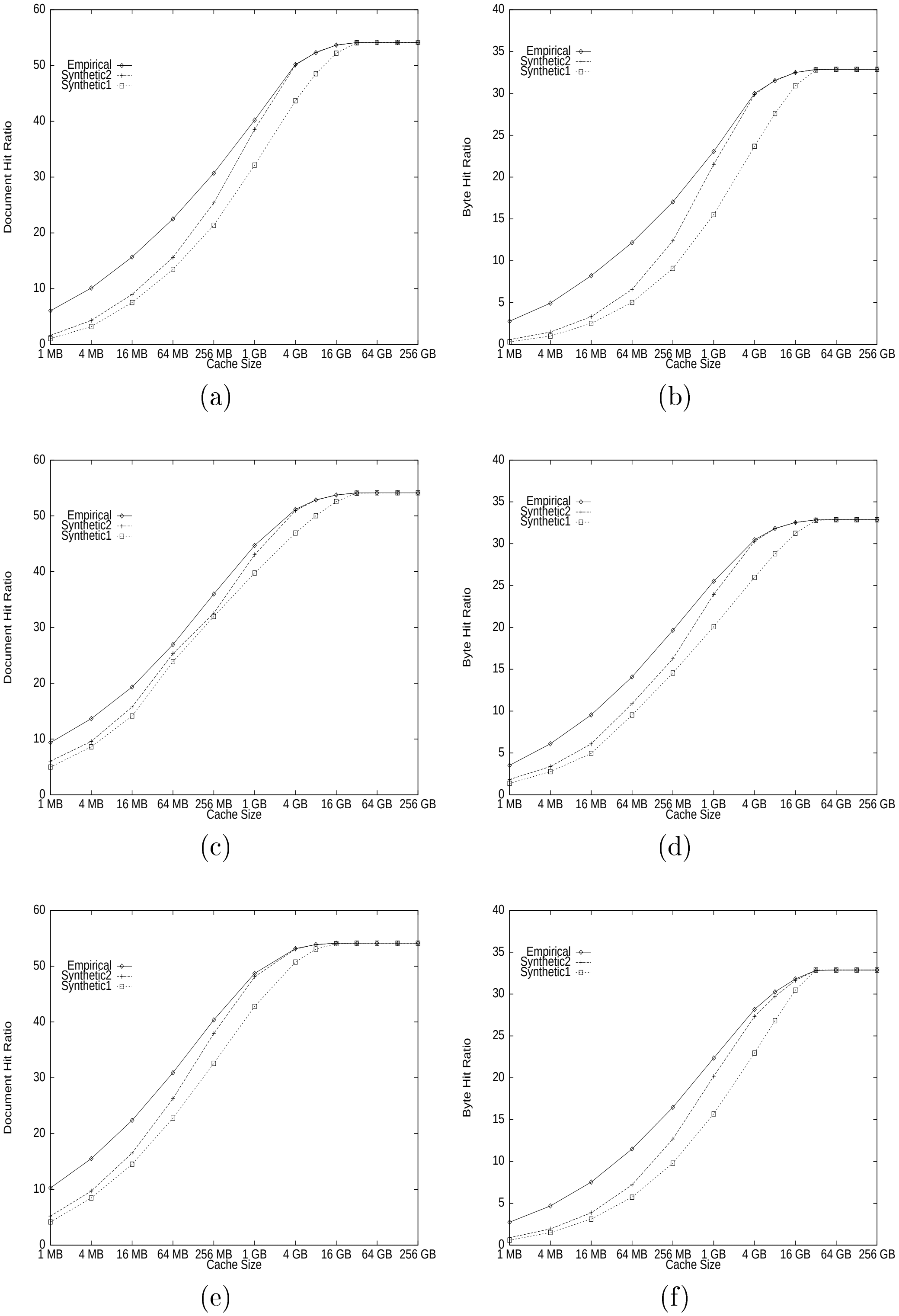
The relative performance of three replacement policies are similar for both USask and NLANR traces. The GD-Size(1) policy achieves higher document hit ratios by favoring smaller documents over large ones, and by aging documents not referenced for a long duration of time. LFU-Aging performs better than LRU, by keeping frequently requested popular documents in the cache and quickly removing infrequently referenced documents. LRU achieved the lowest document hit ratios among the three policies considered. It is also observed that the LFU-Aging policy outperforms both LRU and GD-Size(1) in terms of byte hit ratios. LFU-Aging attains higher byte hit ratios because it retains popular documents in its cache for a longer time and does not discriminate against larger documents. Since GD-Size(1) discriminates against large documents, its performance is the worst among the three replacement policies considered, with respect to byte hit ratio.

The LRU policy is known to work well when there is strong correlation between past and future references. Since LRU performed the worst (in terms of document hit ratio) among the replacement policies considered, perhaps temporal locality is not a major factor in Web proxy workloads. If that were the case, then it would be possible to employ very simple synthetic workload models in Web performance studies, in which document references were modeled as independent random events with stationary probabilities. This question is answered by comparing the hit ratios for the empirical traces with those for the synthetic traces. Figure 4 presents the simulation results for the USask traces. Similar results were obtained for the NLANR traces.

In general, it is observed that the cache hit ratios for the synthetic traces underestimate the hit ratios obtained with the corresponding empirical trace. It is also observed that as the size of the cache increases, the hit ratios obtained by the `Synthetic2` trace get closer to those obtained with the empirical trace. Note that for cache sizes close to the average daily unique document set size (i.e., the average total size of all unique documents accessed in a single day), the hit ratios obtained with the `Synthetic2` trace are close to those obtained for the empirical trace.

Some interesting observations can be made regarding the influence of temporal locality on the performance of the three caching policies considered. First, among the three cache replacement policies considered, the LRU policy shows the greatest sensitivity to the degree of temporal locality present. Second, the LFU-Aging policy is observed to be the least affected by temporal locality. Finally, the impact of temporal locality on GD-Size(1) decreases much more rapidly compared to LRU and LFU-Aging, with an increase in cache size.

These results show that temporal locality is indeed important for analyzing quantitative aspects of Web caching, and the impact of temporal locality varies with the replacement policies. It also establishes that a simple IRM-



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 4. Caching Performance Results for Empirical and Synthetic Traces for the USask Data Set: (a) Document Hit Ratio (LRU); (b) Byte Hit Ratio (LRU); (c) Document Hit Ratio (LFU); (d) Byte Hit Ratio (LFU); (e) Document Hit Ratio (GD-Size); (f) Byte Hit Ratio (GD-Size)

based model is not adequate for Web proxy workload generators designed to assess various qualitative and quantitative aspects of Web proxy cache performance. However, it is important to realize that Web caches are typically many gigabytes in size. In this context, a synthetic trace generated by applying IRM on a day-to-day basis can yield accurate results as the cache performance results achieved for such traces are close to those obtained for the corresponding empirical traces [5] .

6 Source of Temporal Locality

Most modern Web browsers implement some type of memory/disk cache. While one can expect temporal locality to exist at the browser-level because of individual surfing habits, it is also reasonable to expect most repeated references to cacheable documents to be satisfied by the browser caches. However, contrary to intuition, we observe considerable short-term temporal locality at the proxies. Furthermore, in related research [16], one of the authors observed that approximately 30% of the total re-references to documents occur within an inter-reference time of one minute. The presence of short-term locality indicates two possible scenarios: (a) there is much temporally-correlated document sharing among clients; and/or (b) some documents that are deemed uncacheable by the browser caches are considered to be cacheable by the proxy.

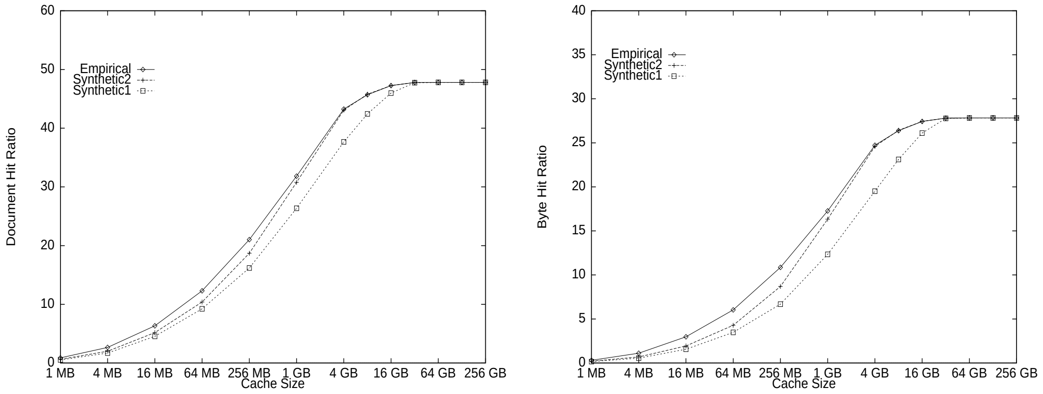
We designed a simple experiment to identify document references one would expect to have been satisfied at the browser level. The simulator was modified to simulate a small cache at each client. The USask empirical trace was replayed through this modified simulator with per-client browser caches of size 1 MB [6] using an LRU replacement policy. All the requests that resulted in a hit at a browser cache were then removed from our empirical trace. Using this reduced data set, the experiments described in Section 5.2 were repeated. The simulation results using LRU as the replacement policy are shown in Figure 5.

The simulation results show that the hit ratios obtained with the synthetic traces are substantially more similar to those obtained with the empirical trace, indicating that much of the short-term locality that was observed in the original empirical trace has been removed. Apparently, then, much of this short-term locality must be owing to clients without local caches, or with caches that have differing notions of cacheablity than do the proxies [7] .

[5] In [16], a model for incorporating short-term locality in the synthetic workloads was proposed and validated.

[6] To be on the conservative side, we assumed very small browser caches.

[7] We are confident that the original empirical trace largely included only references that were considered cacheable at the proxy, owing to the filtering technique



(a) (b)

Fig. 5. Cache Performance Results for the Reduced Empirical versus Synthetic Traces for the USask Data Set using the LRU Replacement Policy (a) Document Hit Ratio; (b) Byte Hit Ratio

7 Conclusions

The first part of this paper analyzed both short-term and long-term temporal locality characteristics observed in Web proxy workloads. A "hot set" drift analysis showed that there are a significant number of documents that have enduring popularity. Then, using a novel measure of temporal locality that distinguishes "hot set" effects from short-term temporal locality, it was established that short-term temporal locality is indeed present in Web proxy workloads.

The second part of this paper considered the impact of temporal locality on Web proxy cache performance using empirical and synthetic traces. Three different cache replacement policies were used: LRU, LFU-Aging and GD-Size(1).

Comparison of the hit ratios for the empirical traces with those for synthetic traces based on the IRM model show that the synthetic traces underestimate the hit ratios. However, for cache sizes larger than the average total size of all unique documents referenced in a single day, short-term temporal locality does not significantly impact performance . For such caches, synthetic traces generated by applying IRM on a day-to-day basis can yield accurate results. Finally, it appears that most of the short-term temporal locality observed at the Web proxies considered in this work is due to clients without browser caches, or with caches that make differing cacheability assumptions than do the proxies.

described at the end of Section 3.

Acknowledgements

Financial support for this research was provided by TR*Labs*, CANARIE, and NSERC Research Grants OGP0120969 and OGP0000264.

The authors thank NLANR and the Department of Computing Services at the University of Saskatchewan for making the proxy access logs available for our study. The authors are grateful to Greg Oster for his valuable assistance in trace collection, trace analysis, and technical support for our project. We also thank Hewlett-Packard for an equipment donation that helped make this project possible.

References

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

[2] M. Arlitt, C. Williamson, Internet Web Servers: Workload Characterization and Performance Implications, IEEE/ACM Transactions on Networking 5 (5) (1997) 631-645.

[3] M. Baker, J. Hartman, M. Kupfer, K. Shirriff, J. Ousterhout, Measurement of a Distributed File System, Proceedings of 13th ACM Symposium on Operating System Principles, Pacific Grove, CA, October 1991, pp. 198-212.

[4] M. Baentsch, L. Baum, G. Molter, S. Rothkugel, P. Sturm, Enhancing the Web's Infrastructure: From Caching to Replication, IEEE Internet Computing 1 (2) (1997) 18-27.

[5] P. Barford, A. Bestavros, A. Bradley, M. Crovella, Changes in Web Client Access Patterns: Characteristics and Caching Implications, World Wide Web Journal 2 (2) (1999) 15-28.

[6] H. Braun, K. Claffy, Web Traffic Characterization: An Assessment of the Impact of Caching Documents from NCSA's Web Server, Computer Networks and ISDN Systems 28 (1-2) (1995) 37-51.

[7] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, On the Implications of Zipf's Law for Web Caching, Proceedings of IEEE INFOCOM'99, New York, March 1999, pp. 126-134.

[8] R. Caceres, F. Douglis, A. Feldmann, G. Glass, M. Rabinovich, Web Proxy Caching: The Devil is in the Details, Performance Evaluation Review 26 (1) (1998) 11-15.

[9] P. Cao, S. Irani, Cost-Aware WWW Proxy Caching Algorithms, Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, Monterey, CA, December 1997, pp. 193-206.

[10] P. Denning, S. Schwartz, Properties of the Working Set Model, Communications of the ACM 15 (3) (1972) 191-198.

[11] B. Duska, D. Marwood, M. Feeley, The Measured Access Characteristics of

World Wide Web Client Proxy Caches, Proceedings of the USENIX Symposium on Internet Technologies & Systems, Monterey, CA, December 1997, pp. 23-35.

[12] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Hypertext Transfer Protocol – HTTP/1.1, HTTP Working Group, Internet Draft, November 1998, available at URL http://www.w3.org/Protocols/HTTP/1.1/draft-ietf-http-v11-spec-rev-06.txt.

[13] S. Glassman, A Caching Relay for the World-Wide Web, Computer Networks and IDSN Systems 27 (2) (1994) 165-174.

[14] T. Kroenger, D. Long, J. Mogul, Exploring the Bounds of Web Latency Reduction from Caching and Prefetching, Proceedings of the USENIX Symposium on Internet Technology & Systems, Montery, CA, December 1997.

[15] A. Luotenen, K. Altis, World-Wide Web Proxies, Computer Networks and ISDN Systems 27 (2) (1994) 147-154.

[16] A. Mahanti, Web Proxy Workload Characterisation and Modelling, M.Sc. Thesis, Department of Computer Science, University of Saskatchewan, September 1999, available at URL ftp://ftp.cs.usask.ca/pub/discus/thesis-mahanti.ps.Z.

[17] A. Mahanti, C. Williamson, D. Eager, Traffic Analysis of a Web Proxy Caching Hierarchy, IEEE Network, Special Issue on Web Performance, 14 (3) (2000) 16-23.

[18] S. Majumdar, R. Bunt, Measurement and Analysis of Locality Phases in File Referencing Behavior, Proceedings of the 1986 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, Raleigh, NC, September 1986, pp. 180-192.

[19] National Laboratory for Applied Network Research, NLANR sanitized access logs, available at URL ftp://ircache.nlanr.net/Traces/.

[20] N. Niclausse, Z. Liu, P. Nain, A New Efficient Caching Policy for WWW, Proceedings of the 1998 Workshop on Internet Server Performance (WISP '98), Madison, WI, June 1998, pp. 119-128.

[21] A. Smith, Cache Memories, Computing Surveys 14 (3) (1982) 473-480.

[22] J. Spirn, Distance String Models for Program Behavior, IEEE Computer 9 (11) (1976) 14-20.

[23] Squid Internet Object Cache, http://squid.nlanr.net/.

[24] D. Wessels, K. Claffy, ICP and the Squid Web Cache, IEEE Journal on Selected Areas in Communication 16 (3) (1998) 345-357.

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

[26] D. Willick, D. Eager, R. Bunt, Disk Cache Replacement Policies for Network File Servers, Proceedings of the 13th International Conference on Distributed Computing Systems (ICDCS), Pittsburgh, PA, May 1993, pp. 2-11.

[27] C. Wills, M. Mikhailov, Examining the Cacheability of User-Requested Web Resources, Proceedings of the 4th International Web Caching Workshop, San Diego, CA, March/April 1999, pp. 78-87.

[28] A. Wolman, G. Voelker, N. Sharma, N. Cardwell, M. Brown, T. Landray, D. Pinnel, A. Levy, Organization-Based Analysis of Web-Object Sharing and Caching, Proceedings of the USENIX Symposium on Internet Technology & Systems, Boulder, Colorado, October 1999.