Locality Characteristics of Web Streams Revisited

Aniket Mahanti   Anirban Mahanti   Carey Williamson

Department of Computer Science, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4
Email: {amahanti, mahanti, carey}@cpsc.ucalgary.ca

***Abstract*— This paper studies *locality of reference* properties of Web streams using a recently proposed Aggregation-Disaggregation-Filtering framework. Two primary research questions are addressed: 1) What impact does locality of reference have on caching performance? and 2) What are the locality characteristics of streams that result from aggregation of filtered streams? Trace-driven simulations are used to answer these questions. The simulation results show which caching policies are adept at exploiting locality characteristics. The results also illustrate the locality properties of the resulting filtered streams.**

I. Introduction

Since its inception in the early 1990s, the Web has experienced phenomenal growth in the number of users, the number of Web servers, and the volume of Internet traffic generated. This growth has resulted in significant structural changes to the Web, aimed at improving its performance and scalability.

Web caching proxies have been widely deployed as a means to reduce network traffic and improve response times for Web accesses. Proxies act as intermediaries between clients and servers, forwarding requests that the proxy is unable to satisfy locally. A proxy is able to satisfy a request locally if the requested document is already present in its cache, resulting in a cache hit. Otherwise, a cache miss occurs and the proxy forwards the request to an appropriate server that returns the document to the proxy. The proxy, in turn, forwards the document to the client and also stores a copy in its cache. Over the years, Web caching has transformed into a multi-level system of interconnected caches, with caches organized either in a mesh or a tree-like hierarchical configuration [7], [10], [25], [31], [35].

Web workload characterization has received considerable attention (e.g., see [4], [13], [21], [25] and the references therein) as such studies offer useful insights into the design and performance of the Web. For example, insight gained from these studies has resulted in the development of better caching policies [9], [19], [29], [30], [34], prefetching techniques [11], and load distribution policies for server/proxy clusters [20]. However, most prior work has focused on analysing Web request streams in *isolation*, without considering the transformations Web streams undergo as they traverse proxy caches.

Fonseca *et al.* [15], [16] recently proposed a general Aggregation-Disaggregation-Filtering (ADF) framework for analysing the transformations of Web streams as they traverse multiple caches. This framework considers three basic transformations, namely *aggregation*, *disaggregation*, and *filtering*. Aggregation refers to multiple streams being merged into a single stream based on their arrival times. A typical example is the aggregation of requests from multiple sources (clients and proxies) at a server. Disaggregation is the reverse of aggregation where a single stream is split into multiple streams based on destination addresses. A typical example is the forwarding of requests by a proxy server to different origin servers. Filtering is a by-product of caching wherein some requests in a stream are absorbed by the proxy as cache hits, while others (those that result in cache misses) are forwarded to a higher-level proxy, or the origin server.

Among the properties of Web workloads, *locality of reference* is one of the most important characteristics affecting caching performance. Locality manifests itself as non-uniform referencing of documents over short-term and long-term time scales. This property can be exploited by cache replacement policies to determine which documents should be kept in the cache and which documents should be evicted, given a fixed-size cache [17]. Cache replacement policies that exploit locality characteristics, however, produce filtered streams that have significantly less locality. This filtering behaviour has been quantified in empirical measurements of Web proxy caching hierarchies [25], where it has been observed that the document hit ratios decreased significantly at higher levels of a proxy caching hierarchy.

The goal of this paper is to study how reference locality influences Web caching performance within the ADF framework. Specifically, the following questions are addressed:

- What impacts do locality characteristics have on caching performance? Here the focus is on determining which caching policies are more adept at exploiting locality characteristics. Also of interest are the locality properties of the resulting filtered streams.
- What are the locality characteristics of streams that result from aggregation of filtered streams? This question helps us understand the advantages/disadvantages of organizing proxies in hierarchies.

Trace-driven simulations are conducted to answer the above questions. Regarding caching performance, we observe that cache replacement policies should attempt to exploit reference locality arising from temporal correlation between document references as well as that arising from sheer popularity of documents. Furthermore, our results indicate that the Greedy Dual-Size [9] replacement policy has robust performance for a wide range of locality characteristics. Finally, our simulation results indicate that in many scenarios, aggregating misses from several child proxies at a parent proxy only marginally increases reference locality. The latter result suggests limited advantages for organizing proxies in caching hierarchies.

The rest of the paper is organized as follows. Section II reviews the principle of locality and the terminology used in the paper. Section III describes the experimental methodology for the simulation study. Section IV discusses the simulation results for filtering at the lower-levels of a Web caching hierarchy. Section V quantifies locality characteristics of reference streams obtained by aggregating filtered streams at a higher-level proxy. Section VI presents conclusions and future work.

II. Principle of Locality

Fundamental properties of locality were first established by Denning and Schwartz in the context of memory systems [14]. Subsequently, locality properties were observed and studied in the context of file referencing behaviour [26], [28], [32], distributed file servers [5], [36], and more recently in the context of Web workloads [3], [4], [13], [16], [18], [25], [33]. For a string of references to a set of objects[†], the principle of locality asserts that: 1) during any interval of time, the references are non-uniformly distributed over the objects; 2) the frequency of reference to any object changes slowly over time; and 3) the correlation between immediate past and immediate future references tends to be high, whereas the correlation between distant references tends to be low [14].

Locality properties have been quantified, to varying extents, in Web workloads. First, Web references are known to be governed by a Zipf or Zipf-like distribution [2], [4], [6], [25]. In the literature, two distinct terms are used to describe this behaviour: *concentration* [25] and *popularity* [18]. Second, empirical studies of Web workloads have established that the set of references to the most popular objects are quasi-stationary [23]. The third locality property, also referred to as *temporal locality* in the literature, has also been widely observed in Web workloads [12], [16], [18], [23]. Previous work has established two sources of temporal locality: correlation between references due to the surfing habits of individual clients, and correlation between document references due to sheer popularity of objects [18], [23].

Understanding locality properties is crucial to the success of demand-driven caching schemes. Cache replacement policies attempt to remove from the cache objects that are unlikely to be referenced soon, based on past reference behaviour. One consequence of a Zipf-like distribution is that a small fraction of the total objects can account for a large fraction of the total references, indicating that frequency-based cache replacement policies might be fairly successful in increasing cache hit rates [6]. Similarly, a stream with a high degree of temporal locality will tend to reference in the near future documents that have been referenced in the recent past, and thus might benefit from the use of a recency-based cache replacement algorithm. In this work, we restrict our attention to popularity and temporal locality characteristics of Web streams, and study the transformations that take place when streams undergo filtering and aggregation.

[†]We use "objects" as a general term to refer to Web documents, files, or memory addresses, as the case may be for Web references, file references, or memory references, respectively.

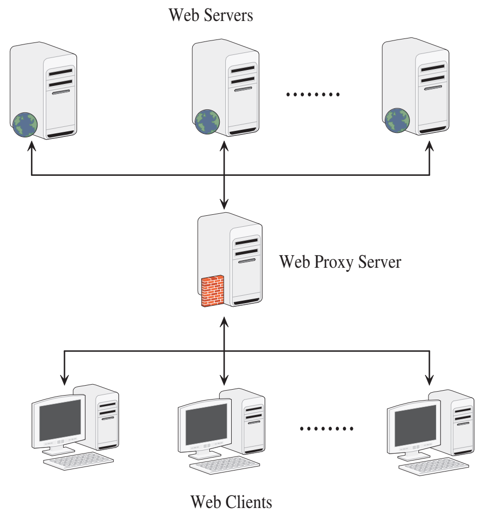


Fig. 1. Single-level Web Proxy Caching Hierarchy Simulation Model

III. Experimental Methodology

This section describes the experimental methodology used for understanding filtering and aggregation effects in single-level and two-level caching hierarchies.

A. Web Proxy Caching Hierarchy Models

Two simulation models are considered in this study.

The first simulation model (see Figure 1) considers a single Web proxy directly receiving requests from several clients. This proxy acts as an intermediary between the clients and the Web servers. This simulation model is used to study cache filtering effects and to analyse the performance of different caching policies.

The second simulation model (see Figure 2) considers a two-level hierarchical Web proxy configuration where requests from clients are directed to lower-level *child* proxies. The misses at the child proxies are forwarded to an upper-level *parent* proxy. If the parent proxy is unable to satisfy a request, it retrieves the document from the server and sends it to the client through the appropriate child proxy. Document caching at each level of the hierarchy is determined by a caching strategy. This simulation model is used to analyse the properties of the aggregated stream as seen by the parent proxy. The experiments consider aggregated streams from two, four, and eight child proxies.

Web caching hierarchies deeper than the two-level configuration considered in this study are the subject of future work. However, there is evidence in the literature that suggests that caching hierarchies with several levels may be undesirable [22], [24], [25], [35]. For example, empirical measurements from a three-level caching hierarchy indicate document hit ratios of 35-40% at a university-level proxy, hit

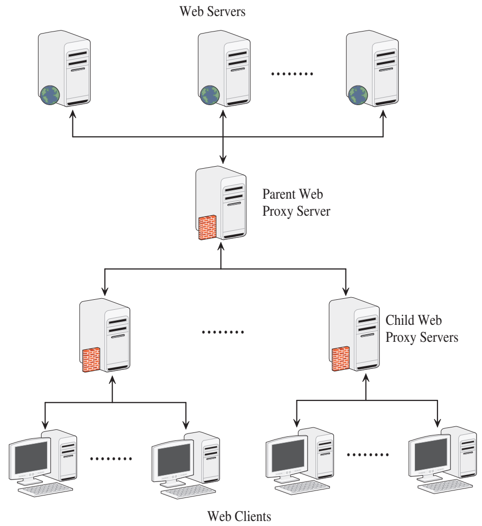


Fig. 2. Two-level Web Proxy Caching Hierarchy Simulation Model

ratios of 15-20% for a national-level proxy, and hit ratios of 5-10% for a root-level NLANR cache [24]. Since the chances of finding a document of interest decreases as the search moves up in the caching hierarchy, caching hierarchies are said to suffer from *diminishing returns* [1], [7], [35]. Furthermore, deeper caching hierarchies may increase document access latencies. This is because cache hits at the higher levels incur longer delays in locating the cached document as well as percolating the document through the hierarchy (compared to a cache hit at a lower level).

B. Factors and Levels

Five factors are considered in the experiments: Web proxy workload, degree of temporal locality, degree of aggregation, cache replacement policy, and cache size.

1) Web Proxy Workload: This study uses synthetic Web proxy traces as they provide a flexible means of controlling the locality properties of the workloads. The synthetic traces are generated using WebTraff [27]. WebTraff is a Web proxy workload generation tool that statistically models five important Web workload characteristics that affect caching performance, namely one-time referencing, document popularity, document size distribution, correlation between document size and popularity, and temporal locality.

In WebTraff, temporal locality is introduced in the traces using the LRU Stack Distance (LRU-SD) model [8], [27]. The LRU-SD model is a stack-based ordering of requests according to their recency of reference with the most recently referenced document located at the top of the stack. Whenever a document is referenced, the LRU stack is searched for the document and if found, the document is removed from its present position in the stack, and moved to the top of the stack while pushing the documents above it down the stack. If a document is not found in the stack, it is simply added to the top of the stack; the other documents in the stack are moved down the stack by one position. Each position in the stack has an associated probability of reference, determined from the analysis of empirical workloads.

TABLE I

CHARACTERISTICS OF SYNTHETIC WEB PROXY WORKLOADS

Property	*Trace1*	*Trace2*
Total requests	1,480,336	1,480,336
Unique documents	495,000	495,000
Unique documents (% of requests)	33	33
One-timers	345,835	345,835
One-timers (% of unique documents)	70	70
Total size of unique documents (GB)	6	6
Total size of trace (GB)	14	14
Smallest document size (bytes)	25	25
Largest document size (bytes)	36,454,766	36,454,766
Median document size (bytes)	3501	3501
Mean document size (bytes)	10,245	10,245
Zipf Slope	0.8443	0.8443
Pareto Tail Index	1.2	1.2
Stack Depth (**Temporal Locality**)	1 (**Low**)	1000 (**High**)

By varying the stack depth used in the workload generator, different degrees of temporal locality can be modelled in the traces. WebTraff incorporates two versions of the LRU-SD model: *static* and *dynamic*. The dynamic model introduces document-specific temporal locality in the workload, while the static model results in homogeneous temporal locality among the documents of the workload. This study uses the *static* LRU-SD model for the experiments.

Table I summarizes the characteristics of the traces used in this paper. These traces differ only in the temporal locality characteristics. *Trace1* has low temporal locality, while *Trace2* has high temporal locality.[‡] Stack depths of 1 and 1000 were used for generating *Trace1* and *Trace2*, respectively.

For the two-level Web proxy model, a separate trace is used as input for each child proxy. An alternative approach is to generate inputs for the child proxies by splitting a big trace into smaller ones. However, our approach provides greater control over the characteristics of the workload and the desired degree of document overlap.

Two workload overlap models are considered, namely a *no overlap* model and a *partial overlap* model. The first model represents a situation where traces observed by the child proxies have no documents in common. The latter model represents the scenario where references to certain documents are observed at all child proxies. In the experiments reported here, 50% of the unique documents seen at a child proxy are common with the other child proxies in the caching hierarchy.

2) Degree of Temporal Locality: Simulation experiments were conducted using traces with varying degrees of temporal locality. Results are reported for *Trace1* and *Trace2*, the traces with the lowest and highest degrees of temporal locality.

3) Degree of Aggregation: The aggregation experiments consider aggregating misses at a parent proxy from two, four, and eight child proxies.

4) Cache Replacement Policies: Cache replacement policies set the criteria for evicting documents resident in the cache

[‡]We also conducted several experiments with traces that have intermediate degrees of temporal locality.

to make room for new documents, when there are constraints on the cache size. Five different cache replacement policies are considered, namely: Least Recently Used (LRU), Least Frequently Used (LFU), Greedy Dual-Size (GDS), Random (RAND), and First In First Out (FIFO) [30]. These policies reflect a broad range of (cache) replacement algorithms found in the literature.

The LRU policy is a recency-based policy that removes from the cache the document that has not been accessed for the longest period of time. This policy has been widely used in various computer systems for many years.

The LFU policy is frequency-based. It keeps a reference count for every document in the cache and removes the document with the lowest count value. The implementation of LFU in the simulations provides aging of documents that build up high reference counts but are not requested again [23], [30]. The aging policy halves the reference counts for all documents in the cache when the average reference count exceeds a specified threshold. The aging mechanism makes documents that have not been referenced for an extended period of time eligible for removal from the cache.

The GDS policy is a size-based policy. It maintains a utility value $H = \frac{1}{s}$ for every document in the cache, where s is the size of the document [9]. The document with the lowest H value is removed from the cache. The H values for all other documents are reduced by the value for the evicted file.

The RAND policy randomly chooses a document for removal from the cache, while the (arrival-based) FIFO policy removes the oldest document in the cache.

5) Cache Size: In the first simulation model, fifteen cache sizes are considered. These are: 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB, 2 GB, 4 GB, 8 GB, and 16 GB. An upper bound of 16 GB is chosen as it reflects an infinite cache size for the synthetic traces considered here. Infinite cache size is useful for determining the maximum achievable cache hit ratios.

For the second simulation model, nine cache sizes are used, namely, 1 MB, 2 MB, 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128 MB, and 256 MB.

C. Performance Metrics

Four performance metrics are used in the simulation study. These are document hit ratio, byte hit ratio, cumulative reference measure, and inter-request measure.

The effectiveness of the caching performance is measured using the hit ratios. Document hit ratio is defined as percentage of total requests that are satisfied by the proxy. Byte hit ratio is the percentage of total volume of data (in bytes) that is satisfied by the proxy. Higher hit ratios signify better performance of the cache replacement policy. A higher document hit ratio would mean that more documents are being filtered at the proxy, and thus less load on the server. A higher byte hit ratio implies lower bandwidth consumption between the proxy and the server (i.e., more requests are satisfied at the proxy itself).

Locality characteristics are measured using two new metrics. The cumulative reference measure of a request stream is defined as the fraction of total requests accounted for by the top 10% of the most popular documents. This measure quantifies popularity. The inter-request measure is the probability of referencing a document again within at most 1000 intervening requests to other documents. This measure quantifies temporal locality.

IV. Simulation Results for Filtering

This section presents results from the trace-based simulation experiments. The simulation results are organized into three sections: Section IV-A summarizes caching performance of the replacement policies, Section IV-B describes the impact of filtering on popularity characteristics, and Section IV-C discusses temporal locality properties of the filtered traces.

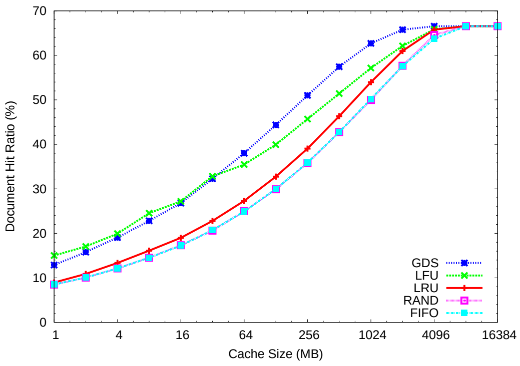
A. Performance of Cache Replacement Policies

Figure 3 depicts the document hit ratios and byte hit ratios for the two workloads. Among the caching policies considered, GDS consistently performs better than the others with respect to document hit ratio. This is because GDS tends to keep smaller documents in the cache (i.e., more documents reside in the cache). The bias of GDS against larger documents, however, results in fewer hits for large documents, limiting the byte hit ratio.

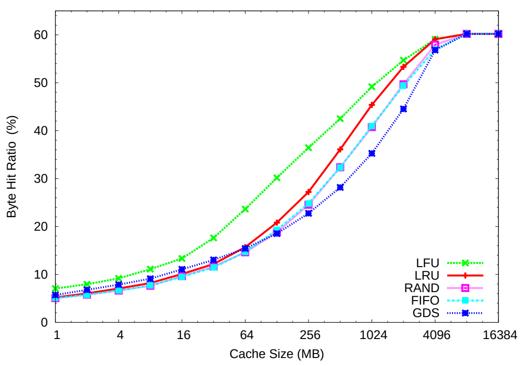
The performance results for RAND and FIFO provide some interesting observations. Both policies have low document hit ratios when the workload has little temporal locality. The low hit ratios occur because these policies do not consider frequency issues when evicting documents from the cache. It might be expected that RAND will perform poorly regardless of the temporal locality characteristics of the traces, since this policy ignores document popularity and temporal locality characteristics. However, the results show that RAND benefits from increased temporal locality in the workload, and even outperforms LFU (see Figures 3(c) and (d)). A FIFO cache naturally captures documents that exhibit temporal locality. As expected, its performance improves with increased temporal locality. These results show that cache replacement policies must consider both types of locality.

The performance of LRU also improves with increased temporal locality. Considering the document hit ratios for a cache size of 16 MB, observe that for *Trace1* (see Figure 3(a)) the absolute difference in the document hit ratios between GDS and LRU is 7.8%. This difference becomes minimal (1.7%) for *Trace2* (see Figure 3(c)). Also note that LRU has the highest byte hit ratio for *Trace2* (see Figure 3(d)). Being a recency-based policy, LRU exploits correlations between document references leading to an improvement in both document and byte hit ratio.

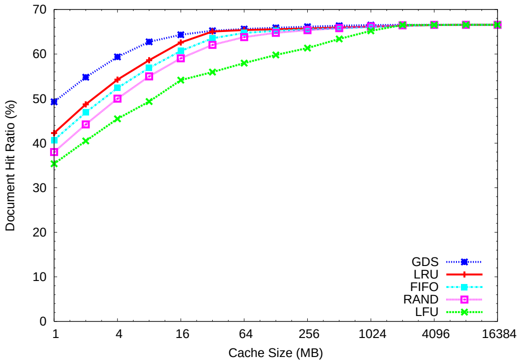
For *Trace2* with stronger temporal locality, the LFU policy generally has the worst hit ratios. A possible explanation of this behaviour is as follows. Note that the LRU-SD model introduces temporal locality in a homogeneous fashion across all documents in the trace. The LFU policy tries to keep the highly referenced documents in its cache. However, it is possible that documents with medium popularity also have



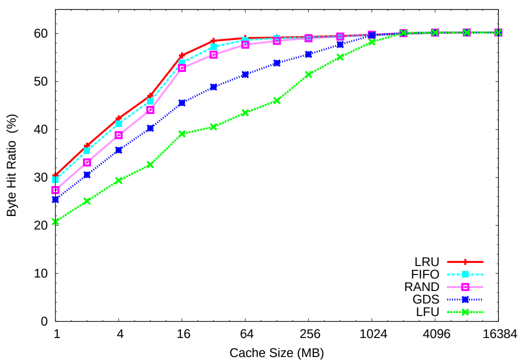
(a) *Trace1*: Document Hit Ratio



(b) *Trace1*: Byte Hit Ratio

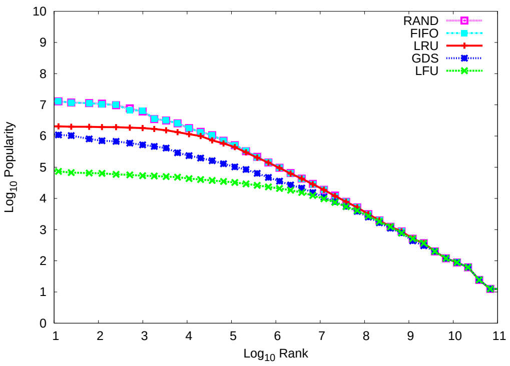


(c) *Trace2*: Document Hit Ratio

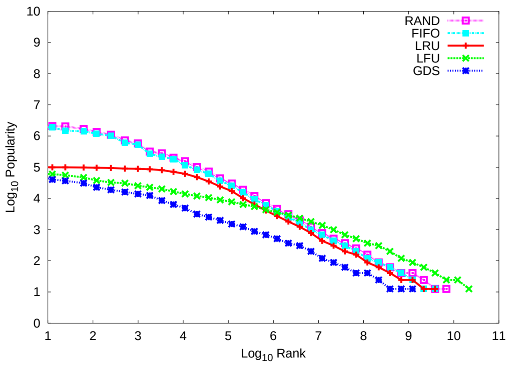


(d) *Trace2*: Byte Hit Ratio

Fig. 3. Document and byte hit ratios for *Trace1* and *Trace2*

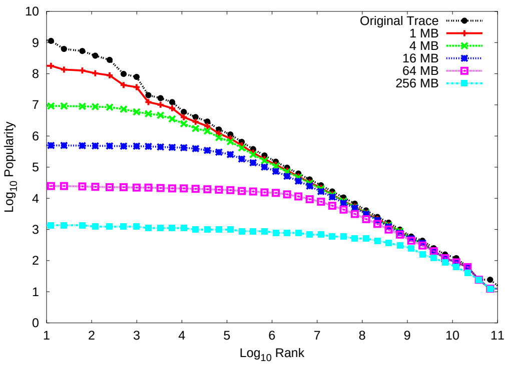


(a) *Trace1*

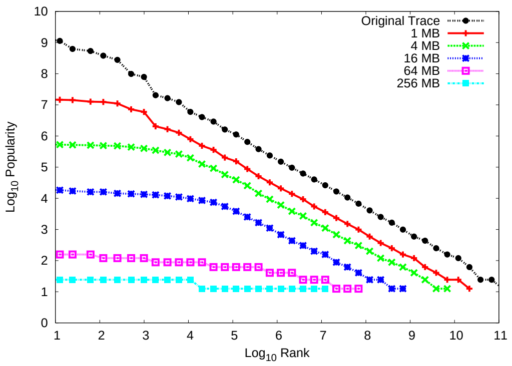


(b) *Trace2*

Fig. 4. Document popularity profile for different cache replacement algorithms (cache size = 8 MB)



(a) *Trace1*



(b) *Trace2*

Fig. 5. Document popularity profile for different cache sizes with the LRU cache replacement policy

temporal locality and thus references to these documents are close together in time. With LFU, these documents with medium popularity could stay in the cache long after their last reference (i.e., before the aging threshold policy makes them eligible for eviction).

B. Popularity after Filtering

Figures 4(a) and (b) show how the popularity characteristics change after cache filtering. These graphs show popularity profile plots of the cache output streams for a cache size of 8 MB, for the five cache replacement policies. The results for *Trace1* and *Trace2* highlight the "flattening effect" for different caching policies. Observe that the popularity profile for GDS is not as flat as that for LRU and LFU. This behaviour can be attributed to the way GDS functions. Since the criteria for evicting documents from the cache is based only on their size, the impact of GDS is felt over a wider range in the popularity profile and not concentrated over the left hand portion of the profile. LFU has the most impact on the popularity profiles of the traces. The flattening effect is more prominent for LFU because this policy keeps the most popular documents in the cache. Policies like RAND and FIFO have little flattening effect on the popularity profile because of their inability to filter popular documents.

Figure 5 demonstrates the filtering effect for different cache sizes and varying degrees of temporal locality. For any given cache size, note the leftward and downward shift of the popularity profile when moving from the results for *Trace1* to those for *Trace2*. The shift can be attributed to increased filtering; as temporal locality increases more requests are absorbed in the cache, and cache misses typically consist of documents with lower popularity. This phenomenon is depicted in the graph by the simultaneous decrease in the height and the length of the popularity profile lines.

Figure 6 shows the cumulative reference measure for the different caching policies versus cache size. Recall that the cumulative reference measure quantifies document popularity in the traces as the percentage of total requests accounted for by the most popular 10% of the documents (i.e., the left-hand portion of the popularity profile). GDS consistently has the greatest reduction in document popularity. This can be attributed to the workload size distribution (i.e., many small documents and a few very large documents). Since GDS is biased towards smaller documents, this policy filters more documents, reducing the overall document popularity.

C. Temporal Locality after Filtering

Figure 7 shows the inter-request measures for the traces. The inter-request measure remains steady for small cache sizes and drops toward zero for larger cache sizes, since a huge cache will satisfy all re-references to a document. The results show that LFU produces the least reduction in temporal locality. Intuitively, this makes sense since LFU only exploits the popularity component of reference locality. It is also interesting to observe that RAND has a higher impact on the inter-request measure than LFU. The decrease in the inter-request measure is most pronounced for the FIFO and LRU policies, both of which exploit temporal locality in the traces. For FIFO, however, exploiting temporal locality alone does not necessarily result in a better document hit ratio.

V. Simulation Results for Aggregation

This section presents selected results from the second set of experiments. The purpose of these experiments is to quantify locality characteristics of reference streams obtained by aggregating filtered cache output traces from the child proxies.

The experiments consider two ($N = 2$), four ($N = 4$), and eight ($N = 8$) child proxies for the purpose of aggregation at the parent proxy. The results for $N = 1$ provide a baseline, representing the trivial case for aggregation. Results are presented for the case where all child proxies run the LRU replacement policy. For $N \geq 2$, both the partial and no overlap workload model as discussed in Section III are considered.

A. Popularity after Aggregation

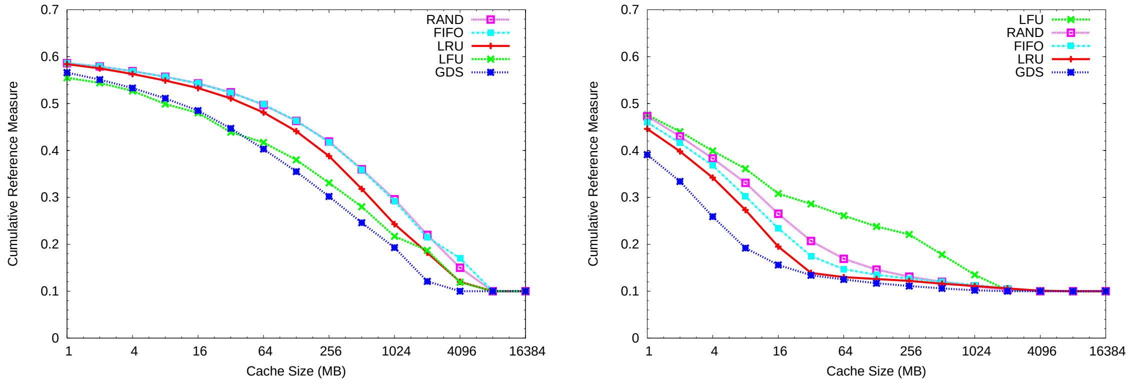
Figure 8 shows the impact of aggregation on popularity under no overlap and partial overlap situations. In Figure 8(a), the cumulative reference results for all N values are the same. This is a characteristics of the no overlap scenario. In this case, the streams entering the child proxies have exactly the same characteristics, but no documents in common. Because the original streams have identical characteristics, the misses from the proxies also have identical characteristics. Therefore, aggregating the misses from N child proxies has the effect of scaling the number of requests, the number of unique documents, and the number of references accounted for by the 10% most popular documents by a factor of N.

In the partial overlap scenario in Figures 8(b) and (c), a modest increase in popularity of the aggregated stream is observed as N increases. Since the request streams at the child proxies have some documents in common, the misses of these streams also have some documents in common. When these miss streams are aggregated, the new stream has a higher concentration of documents that are common across the original streams. This effect increases the number of requests for the popular documents, and thus the cumulative reference measure.

The results presented here all assume the LRU replacement policy at the child proxies. Qualitatively similar results are observed for the other cache replacement policies considered.

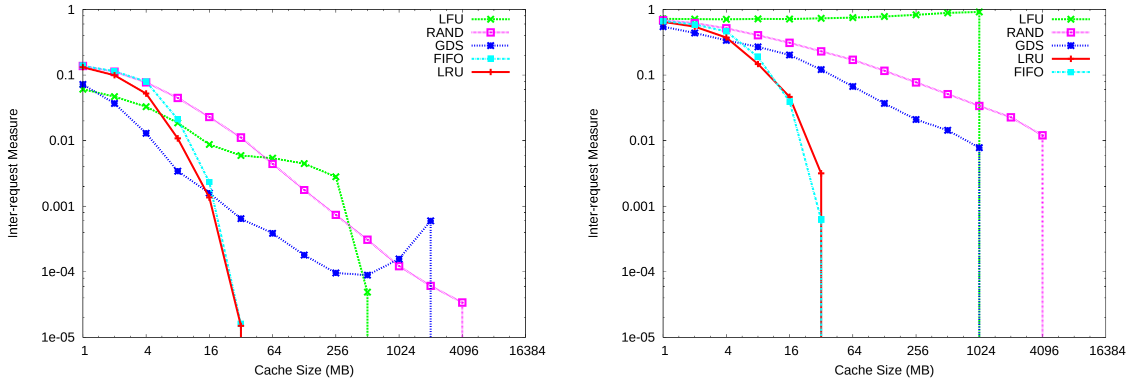
B. Temporal Locality after Aggregation

Figure 9 shows the inter-request measures for the aggregated streams with the LRU policy at the child proxies. Figures 9(a) and (b) show the inter-request measure results when there are no documents common among the reference streams entering the child proxies. Figure 9(a) is for the stream with low temporal locality, while Figure 9(b) is for the stream with high temporal locality. Figures 9(c) and (d) show the inter-request measure results for the partial overlap scenario, when 50% of the documents are common among all reference streams entering the child proxies.



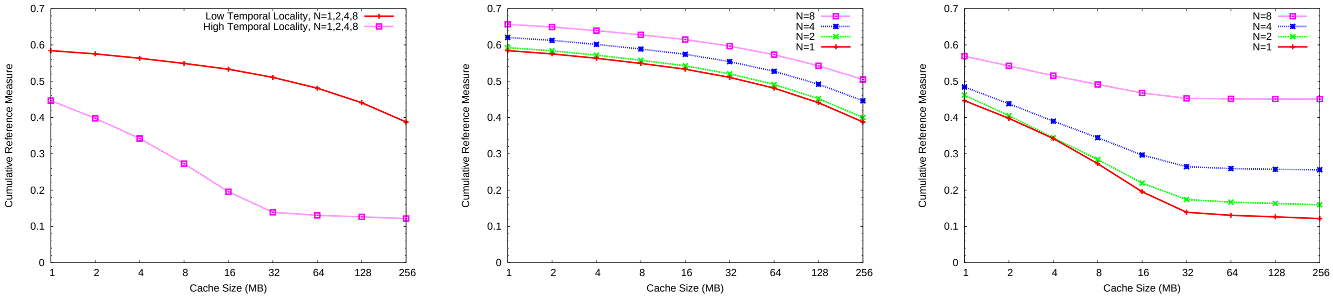
(a) *Trace1* (b) *Trace2*

Fig. 6. Fraction of total requests accounted for by the top 10% of the popular documents for different cache sizes in the filtered stream



(a) *Trace1* (b) *Trace2*

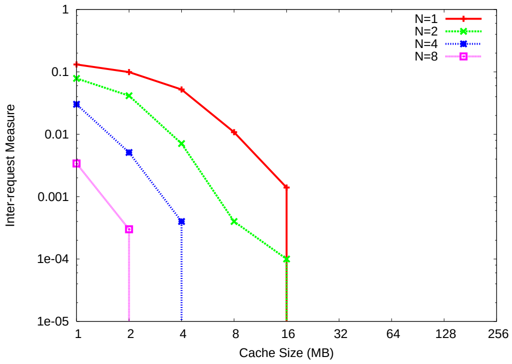
Fig. 7. Probability of re-referencing a document within at most 1000 intervening requests to other documents for different cache sizes in the filtered stream



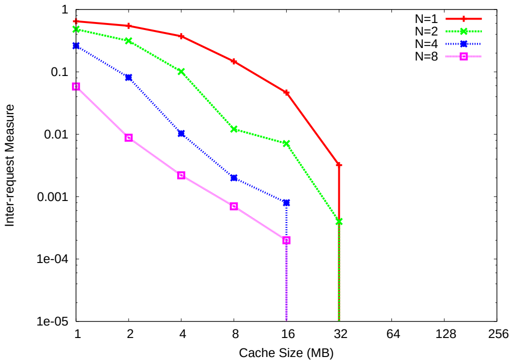
(a) No Overlap (Low/High Temporal Locality) (b) Partial Overlap (Low Temporal Locality) (c) Partial Overlap (High Temporal Locality)

Fig. 8. Fraction of total requests accounted for by the top 10% of the popular documents in the aggregated misses (child proxies running LRU cache replacement policy)

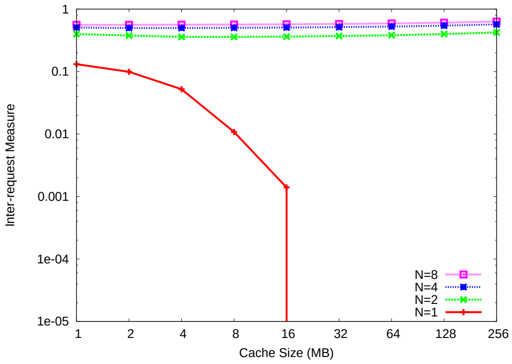
In Figure 9(a), the temporal locality of the aggregated stream decreases as N increases. For example, at a cache size of 1 MB, the inter-request measure for $N = 2$ is reduced by about 60% in comparison to $N = 1$. A similar observation applies for $N = 4$: the inter-request measure is 38% lower than that for $N = 2$. This phenomenon remains consistent over other cache sizes as well and is best explained through an example with $N = 2$. Suppose in the filtered stream from one child proxy a document, say A_1^1, is referenced again after 50 references to other documents, as shown in this string: $A_1^1, U_1^1, U_2^1, \cdots, U_{50}^1, A_1^1$. The design of the no overlap scenario is such that the other child proxy will observe the same situation, albeit with different documents (one that does not exist in the filtered reference stream of the first child proxy), as shown, for example by this string: $A_1^2, U_1^2, U_2^2, \cdots, U_{50}^2, A_1^2$. Note that time stamps (by design) are identical in both reference streams. Thus, the aggregated stream $A_1^1, A_1^2, U_1^1, U_1^2, U_2^1, U_2^2, \cdots, U_{50}^1, U_{50}^2, A_1^1, A_1^2$



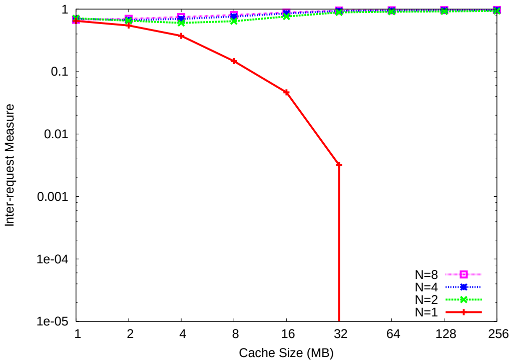
(a) No Overlap: Low Temporal Locality



(b) No Overlap: High Temporal Locality



(c) Partial Overlap: Low Temporal Locality



(d) Partial Overlap: High Temporal Locality

Fig. 9. Probability of re-referencing a document within at most 1000 intervening requests to other documents in the aggregated misses (child proxies running LRU cache replacement policy)

has twice as many documents ($\approx$ 100) between re-references for A_1^1 and A_1^2.

In Figure 9(b), the temporal locality of the aggregated stream also decreases as N increases, though the difference is not as pronounced as in Figure 9(a). In this case, the streams that are directed to the child proxies have a higher degree of temporal locality (i.e., the inter-reference distances are smaller). Thus, even with no documents common amongst the streams, there is enough temporal locality present in the miss streams that its effect is still evident when aggregated.

For the partial overlap scenario, temporal locality in the aggregated stream actually *increases* as N increases. Figures 9(c) and (d) graphically illustrate this phenomenon. This phenomenon can be attributed to the 50% document overlap among all the traces. As an illustration, consider $N = 4$ child proxies and assume that documents A and B (exhibit temporal locality and) are common in a section of the misses of each proxy. A section of concurrent misses from the child proxies might be as follows:

Child Proxy 1:	$A, B, U_1^1 \ldots U_{50}^1, A, B$
Child Proxy 2:	$A, B, U_1^2 \ldots U_{50}^2, A, B$
Child Proxy 3:	$A, B, U_1^3 \ldots U_{50}^3, A, B$
Child Proxy 4:	$A, B, U_1^4 \ldots U_{50}^4, A, B$

Aggregation of the misses at the parent proxy produces the stream: $A, A, A, A, B, B, B, B, U_1^1, U_1^2, U_1^3, U_1^4 \ldots U_{50}^1, U_{50}^2, U_{50}^3, U_{50}^4, A, A, A, A, B, B, B, B$. In this aggregated stream, observe that the references to document A from child proxy 1 are separated by approximately 200 references to other documents. However, the very fact that this document A was present in the streams of the other proxies has resulted in four references to it being clustered together, effectively increasing temporal locality.

VI. Conclusions and Future Work

This paper used trace-driven simulations of synthetic Web proxy workloads to study the impact that locality of reference has on caching performance. Two simulation models were used to understand the filtering and aggregation effects on locality characteristics (popularity and temporal locality) in a single-level and two-level Web proxy hierarchy, respectively.

The first set of experiments showed that the GDS policy consistently performed better than the other cache replacement policies with respect to document hit ratio. The performance of GDS is relatively insensitive to changes in the degree of temporal locality. FIFO and LRU were most successful in exploiting temporal locality in the filtered stream. Also, the hit ratios for LRU increased with an increase in temporal locality. For FIFO, exploiting temporal locality alone does not necessarily result in a better document hit ratio. Both popularity and temporal locality must be considered when designing a good cache replacement policy.

The second set of experiments quantified locality characteristics of reference streams obtained after aggregating filtered cache output streams from the lower-level of the Web caching hierarchy. It was observed that document popularity remained constant in no overlap situations irrespective of the caching policy used. Also, the structural change in the temporal locality of the aggregated stream with increasing number of child proxies is strongly dependent on the degree of overlap among the input streams. This characteristic will ultimately determine the effectiveness of Web proxy caching hierarchies.

Future work will extend this work to deeper caching hierarchies, consider alternative cache organization (e.g., mesh-like organization), and study the impact of heterogeneous cache replacement policies within the hierarchy. We also plan to validate our observations with empirical measurements.

ACKNOWLEDGEMENTS

This research was partially funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada and the Informatics Circle of Research Excellence (iCORE) of the province of Alberta.

The authors also thank the anonymous reviewers for their comments and suggestions which helped improve the paper.

REFERENCES

[1] G. Abdulla, E. Fox, M. Abrams, and S. Williams. WWW Proxy Traffic Characterization with Application to Caching. Technical Report, Virginia Polytechnic Institute and State University, March 1997. Available at: `http://eprints.cs.vt.edu:8000/archive/00000460/`.

[2] L. Adamic and B. Huberman. Zipf's Law and the Internet. *Glottometrics*, 3:143–150, 2002.

[3] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira. Characterizing Reference Locality in the WWW. In *Proc. of the IEEE Conference on Parallel and Distributed Information Systems*, pages 92–103, December 1996.

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

[5] M. Baker, J. Hartman, M. Kupfer, K. Shirriff, and J. Ousterhout. Measurement of a Distributed File System. In *Proc. of the ACM Symposium on Operating Systems Principles*, pages 198–212, October 1991.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. of the IEEE Infocom Conference*, pages 126–134, March 1999.

[7] M. Busari and C. Williamson. Simulation Evaluation of a Heterogeneous Web Proxy Caching Hierarchy. In *Proc. of the IEEE Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 379–388, October 2001.

[8] M. Busari and C. Williamson. ProWGen: A Synthetic Workload Generation Tool for Simulation Evaluation of Web Proxy Caches. *Computer Networks*, 38(6):779–794, 2002.

[9] P. Cao and S. Irani. Cost-Aware WWW Proxy Caching Algorithms. In *Proc. of the USENIX Symposium on Internet Technologies and Systems*, pages 193–206, December 1997.

[10] A. Chankhunthod, P. Danzig, C. Neerdaels, M. Schwartz, and K. Worrell. A Hierarchical Internet Object Cache. In *Proc. of the USENIX Technical Conference*, pages 153–163, January 1996.

[11] X. Chen and X. Zhang. Coordinated Data Prefetching by Utilizing Reference Information at both Proxy and Web Servers. *ACM SIGMETRICS Performance Evaluation Review*, 29(2):32–38, 2001.

[12] L. Cherkasova and G. Ciardo. Characterizing Temporal Locality and its Impact on Web Server Performance. In *Proc. of the Conference on Computer Communications and Networks*, pages 434–441, October 2000.

[13] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of WWW Client-based Traces. Technical Report, Boston University, April 1996. Available at: `http://cs-www.bu.edu/faculty/crovella/paper-archive/TR-95-010/paper.html`.

[14] P. Denning and S. Schwartz. Properties of the Working Set Model. *Communications of the ACM*, 15(3):191–198, March 1972.

[15] R. Fonseca, V. Almeida, and M. Crovella. Locality in a Web of Streams. *Communications of the ACM*, 48(1):82–88, January 2005.

[16] R. Fonseca, V. Almeida, M. Crovella, and B. Abrahão. On the Intrinsic Locality Properties of Web Reference Streams. In *Proc. of the IEEE Infocom Conference*, pages 448–458, April 2003.

[17] A. Foong, Y. Hu, and D. Heisey. Web Caching: Locality of References Revisited. In *Proc. of the IEEE Conference on Networks*, pages 81–86, September 2000.

[18] S. Jin and A. Bestavros. Sources and Characteristics of Web Temporal Locality. In *Proc. of the IEEE Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 28–35, October 2000.

[19] S. Jin and A. Bestavros. GreedyDual* Web Caching Algorithm: Exploiting the Two Sources of Temporal Locality in Web Request Streams. *Computer Communications*, 24(2):174–183, 2001.

[20] M. Kaiser, K. Tsui, and J. Liu. Self-organized Autonomous Web Proxies. In *Proc. of the Conference on Autonomous Agents and Multiagent Systems*, pages 1397–1404, July 2002.

[21] J. Lacort, A. Pont, J. Gil, and J. Sahuquillo. A Comprehensive Web Workload Characterization. In *Proc. of the Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, July 2004.

[22] A. Mahanti. Web Proxy Workload Characterization and Modelling. MSc. Thesis, University of Saskatchewan, Department of Computer Science, September 1999. Available at: `http://pages.cpsc.ucalgary.ca/~mahanti/papers/mscthesis.pdf`.

[23] A. Mahanti, D. Eager, and C. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation*, 42(2-3):187–203, 2000.

[24] A. Mahanti and C. Williamson. Web Proxy Workload Characterization. Technical Report, University of Saskatchewan, March 1999. Available at: `http://pages.cpsc.ucalgary.ca/~mahanti/papers/workloadstudy.pdf`.

[25] A. Mahanti, C. Williamson, and D. Eager. Traffic Analysis of a Web Proxy Caching Hierarchy. *IEEE Network*, 14(3):16–23, 2000.

[26] S. Majumdar and R. Bunt. Measurement and Analysis of Locality Phases in File Referencing Behaviour. In *Proc. of the ACM SIGMETRICS Conference*, pages 180–192, September 1986.

[27] N. Markatchev and C. Williamson. WebTraff: A GUI for Web Proxy Cache Workload Modelling and Analysis. In *Proc. of the IEEE Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 356–363, October 2002.

[28] V. Phalke and B. Gopinath. An Inter-Reference Gap Model for Temporal Locality in Program Behaviour. In *Proc. of the ACM SIGMETRICS Conference*, pages 291–300, 1995.

[29] S. Podlipnig and L. Böszörmenyi. A Survey of Web Cache Replacement Strategies. *ACM Computing Surveys*, 35(4):374–398, 2003.

[30] L. Rizzo and L. Vicisano. Replacement Policies for a Proxy Cache. *IEEE/ACM Transactions on Networking*, 8(2):158–170, 2000.

[31] P. Rodriguez, C. Spanner, and E. Biersack. Analysis of Web Caching Architectures: Hierarchical and Distributed Caching. *IEEE/ACM Transactions on Networking*, 9(4):404–418, 2001.

[32] A. Smith. Cache Memories. *ACM Computing Surveys*, 14(3):473–480, March 1982.

[33] S. Vanichpun and A. Makowski. The Output of a Cache under the Independent Reference Model - Where did the Locality of Reference Go? In *Proc. of the ACM SIGMETRICS Conference*, pages 295–306, June 2004.

[34] J. Wang. A Survey of Web Caching Schemes for the Internet. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46, 1999.

[35] C. Williamson. On Filter Effects in Web Caching Hierarchies. *ACM Transactions on Internet Technology*, 2(1):47–77, 2002.

[36] D. Willick, D. Eager, and R. Bunt. Cache Replacement Policies for Network File Servers. In *Proc. of the Conference on Distributed Computing Systems*, pages 2–11, May 1993.