Time-Domain Analysis of Web Cache Filter Effects

Guangwei Bai **Carey Williamson**
Department of Computer Science
University of Calgary
Calgary, AB, Canada T2N 1N4
{bai,carey}@cpsc.ucalgary.ca

Keywords: Internet and WWW technology, Web proxy caching, workload characterization, traffic modeling

Abstract

This paper uses trace-driven simulation to study the traffic arrival process for Web workloads both before and after a Web proxy cache. In particular, the simulation experiments quantify the filter effects of a Web cache on the request arrival process. Both empirical and synthetic Web proxy workloads are used in the study. The simulation results show that (as expected) a Web cache reduces both the mean and the peak arrival rate for Web traffic workloads. However, the presence of the cache has less effect on the variability of the workload, and no impact on the degree of self-similarity in a workload. Finally, we find that a Gamma distribution provides a flexible and robust means of characterizing the request arrival count distribution, both before and after a Web cache, though the parameters for the Gamma distribution depend upon the input Web workload characteristics and the cache parameters used. This model can be used to estimate traffic characteristics in distributed or hierarchical Web caching architectures.

INTRODUCTION

The World Wide Web (WWW, or the Web) continues to be the major driving force behind the growth in popularity of the Internet. The Web has become the preferred means for the timely dissemination of information in many contexts, including research, education, news, marketing, travel, business, and entertainment domains.

The explosive growth of the Web, with its corresponding increase in Internet traffic volume, has led to user-perceivable network performance problems. In some cases, the performance bottlenecks are at the Web server, if the server architecture cannot handle the client demand. In other cases, the bottleneck is within the Internet itself; network congestion leads to queuing delays and packet losses that degrade Web performance. In yet other cases, excessive delays are due to inefficiencies in the Internet protocol stack (e.g., the interactions between HTTP and TCP), round trip delays across the Internet, or limited client access bandwidth to the Internet.

Web document caching (Web caching) architectures and content distribution networks (CDNs) are now widely used to alleviate these performance problems. By storing copies of popular Web documents close to the users requesting them, Web caches can reduce Web server load and can reduce the volume of traffic traversing the core of the Internet. In many cases, users perceive improved response times for document downloads.

The presence of a cache has a *filter effect* on Web workloads. Because a percentage of the incoming client requests are satisfied directly at the Web cache (i.e., cache hits), a subset of the requests is removed (filtered) from the request workload progressing upstream to the Internet, to other caches, or to the origin servers. This filtering effect is illustrated conceptually in Figure 1, where the original aggregate client workload from an organization traverses Link1 to the organization's Web proxy cache, and the filtered workload traverses Link2 en route to the Internet. The filtering effect in turn changes the response traffic volume traversing Link2 en route to the clients.

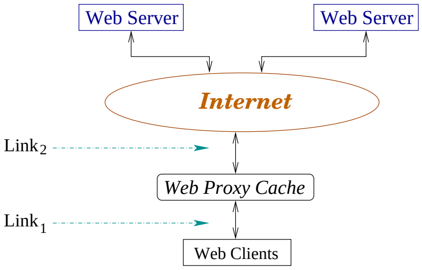


Figure 1. Illustration of Web Cache Filtering Effect

The cache filtering effect manifests itself in two orthogonal ways. First, the cache generally filters out requests for the most popular Web documents (depending of course, on the cache management policy). As a result, the number of HTTP requests for certain Web documents is substantially reduced by the presence of the cache [15]. Second, the presence of the cache changes the structure of the request arrival process entering the Internet, compared to the request arrival process presented to the cache itself.

The first of these two effects has been fairly well studied in the literature [9, 14, 15]. We refer to this work as *frequency-domain* analysis of cache filter effects, since it focuses largely on the frequency distribution of Web document popularity. This document popularity profile typically has a power-law structure before the cache, often characterized using a Zipf or a Zipf-like distribution [4, 5, 12]. After the cache, the "high popularity" end of the Zipf distribution is significantly flattened [9, 15].

To the best of our knowledge, relatively little research work has focused on the second aspect of cache filter effects. Clearly, the presence of the cache reduces the average arrival rate of requests entering the Internet. Indeed, this is the primary motivation for many organizations to install a proxy cache in the first place. In most cases, the cache reduces the peak arrival rate as well, though not necessarily in the same proportion as the mean. However, the impact on the variability (burstiness) of the traffic and the self-similar arrival process [8] is not clear.

The purpose of this paper is to study cache filter effects on the traffic arrival process. We refer to our work as *time-domain* analysis of cache filter effects, to distinguish it from the former frequency-domain effect. Our research is carried out using trace-driven simulations, with empirical and synthetic Web proxy workloads, along with tools for Web caching simulation and traffic characterization and modeling.

The research questions addressed in this paper are:

- What impact does the presence of a Web proxy cache have on the structural characteristics (i.e., mean, peak, variance, self-similarity) of a Web request workload?
- How sensitive is the filter effect to the cache size and the cache replacement policy used?
- How sensitive is the filter effect to the characteristics of the incoming Web workload (i.e., Zipf slope, self-similarity)?
- Can a closed-form mathematical model adequately characterize the cache filter effect?

Our results show that a Web cache is effective in reducing both the mean and the peak arrival rate for Web traffic workloads. The presence of the cache seems to have no impact on the degree of self-similarity in a workload, though in typical cases the filtering effect reduces the relative variance of the outbound request stream. Finally, we find that a Gamma distribution provides a simple, flexibile, and relatively robust means of characterizing the request arrival count process, both before and after a Web cache. The parameters for fitting the Gamma distribution can be estimated from empirical traffic traces, though the fitted parameters for the filtered request stream are strongly dependent on the cache size and the characteristics of the input Web workload. The cache replacement policy has relatively little impact on the traffic structure.

The remainder of this paper is organized as follows. The next section discusses related work on Web workload characterization and cache filter effects, while the section after that describes the empirical Web proxy workload used for our study. The main section of the paper then focuses on understanding Web cache filter effects on the request arrival process, using empirical and synthetic Web traffic workloads. A modeling section follows, in which we propose and validate a parameterizable model for characterizing Web request streams, both before and after a cache. Finally, the paper concludes with a summary of our observations and suggested directions for future research.

BACKGROUND AND RELATED WORK

Web Workload Characterization

Several Web workload characterization studies have appeared in the literature. These studies have focussed on Web client [3], Web server [2], and Web proxy workload characteristics [1, 12].

From these empirical studies, several common workload characteristics emerge that are relevant to Web caching performance. These characteristics include a high degree of *one-time referencing*, a *Zipf-like document popularity distribution*, *heavy-tailed file and transfer size distributions*, and a *temporal locality property* in the document referencing behaviour. These characteristics are quite well-documented in the literature, and are thus not discussed at length here.

Among these characteristics, the one that is most relevant to Web caching performance is the slope of the Zipf-like document popularity distribution [5]. Zipf's law expresses a power-law relationship between the popularity P of an item (i.e., its frequency of reference) and its rank r (i.e., relative rank among the referenced items, based on frequency of reference). This relationship is of the form $P = c/r^{\beta}$, where c is a constant, and β is often close to 1. For example, the frequency of usage for English words in written prose typically follows this distribution.

In the Web context, a similar referencing behaviour is observed [4, 12, 13]. Some researchers have found that the value of the exponent β is close to 1 [1, 3], precisely following Zipf's law. Others [1, 4, 12] have found that the value of β is less than 1, and that the distribution can be described only as "Zipf-like", with the value of β varying from trace to trace. In general, the steeper the Zipf slope, the higher the cache hit ratio achievable for a given Web workload [4, 5, 13].

Cache Filter Effects

Several recent research papers have explored the relationships between Web workload characteristics and Web proxy caching performance [4, 5, 6, 7, 9, 12]. Three of these papers [6, 7, 9] have explicitly addressed the issue of Web cache filter effects, wherein a higher-level cache in a multi-cache system only handles requests that miss in the lower-level cache(s). Similar cache design problems have been addressed previously in the context of CPU cache hierarchies [14], databases [10], and client-server systems [16].

Among the papers that focus on Web cache filter effects, most focus on the “frequency domain” aspect of the Web cache filter effect. Doyle *et al.* [9] refer to this as the “trickle down” effect, and conduct a detailed simulation study to quantify its impact. Che *et al.* [7] propose a frequency-based caching hierarchy, where the lower-level cache handles requests for high-frequency items, and the higher-level cache handles requests for low-frequency items. Busari and Williamson [6] propose a “heterogeneous” Web proxy caching hierarchy that uses different caching policies at different levels of a caching hierarchy. None of these papers explicitly address the structural changes in the request arrival process due to the presence of the Web cache.

WEB PROXY WORKLOAD ANALYSIS

This section describes the empirical Web proxy workload used in our study.

Overview of Empirical Web Proxy Workload

The Web proxy workload used in our study was collected from a campus-wide Web proxy server at the University of Saskatchewan. In this paper, only a one-day access log is used as a representative example of the proxy server workload. This access log was collected on October 17, 2001. This is the same proxy server for which long-duration (9-month) traces were analyzed in previous research [12].

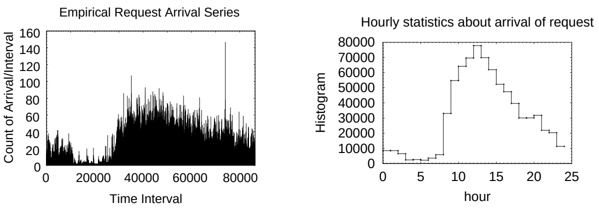
This empirical trace represents a typical one-day workload, from midnight of one day to midnight of the next. The trace contains about 750,000 requests, with request timestamps recorded at a 0.001 second time granularity. Table 1 summarizes the characteristics of the workload.

Figure 2 shows two time series plots illustrating the characteristics of this trace. The horizontal axis shows the time of day, from midnight of one day to midnight of the next. The vertical axis shows the count process for the number of Web requests arriving in each sampling interval (1 second intervals in Figure 2(a), and 1 hour intervals in Figure 2(b)) throughout the day. Figure 2(a) shows that the arrival process is quite bursty throughout the day.

Figure 2(a) shows significant non-stationarities in the daily traffic, a behaviour that is even more evident in Figure 2(b). It is well known that Internet traffic exhibits a daily cyclic pattern, based on the “working hours” for human users. Since this workload is from a university proxy cache, most of the daily Web traffic (71%) occurs between 9am and 6pm.

Table 1. Characteristics of Empirical Web Proxy Workload (University of Saskatchewan Proxy)

Item	Value
Trace Duration	1 day
Trace Date	Oct 17, 2001
Total Requests	755,505
Total Transferred Bytes (Mbytes)	1,087
Mean Transfer Size (bytes)	1,508
Median Transfer Size (bytes)	210
Total Documents	271,285
Unique Documents (% of requests)	35.9%
Total Bytes of Documents (Mbytes)	523
Smallest Document Size (bytes)	0
Largest Document Size (bytes)	86,399,329
Mean Document Size (bytes)	2,021
Median Document Size (bytes)	288
One-timer Documents	201,674
One-timers (% of unique documents)	74.3%
Zipf Slope	-0.8



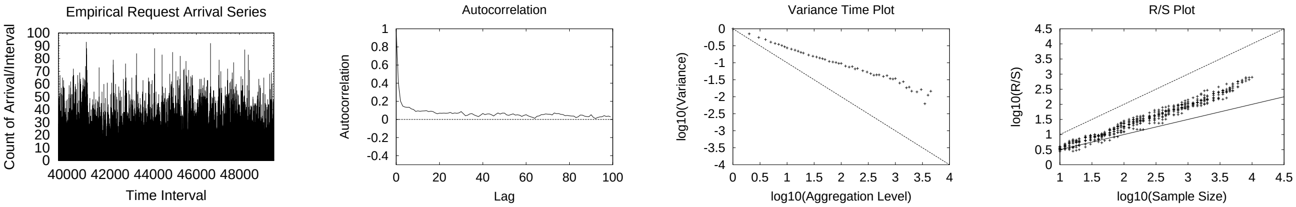
(a) Interval Size: 1 second (b) Interval Size: 1 hour

Figure 2. Time Series Plot of Request Arrival Count Process for Empirical Web Proxy Workload

Self-Similar Arrival Process

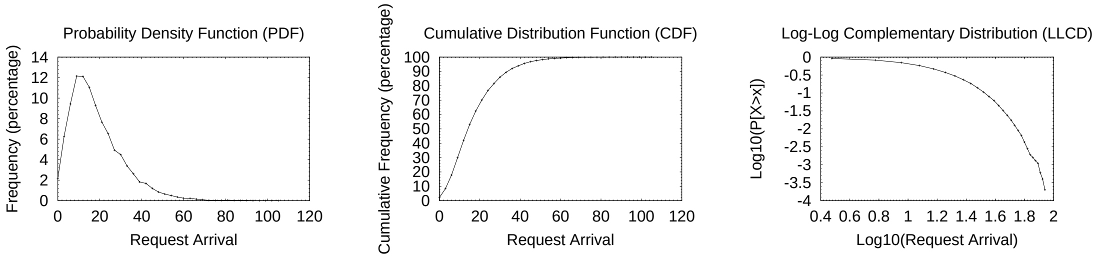
The remaining analyses in this paper focus on the three-hour “busy period” of the trace from 11am to 2pm. This period contains 217,159 requests, representing about 30% of the daily Web request traffic. For this period, we hypothesize that the arrival count process (for 1 second intervals) is stationary (see Figure 3(a)). We use this portion of the trace to characterize the request arrival process, and to test for self-similarity [11] (i.e., long-range dependence) in the arrival count process. We use the standard statistical analysis techniques proposed by Leland et al. [11], namely the autocorrelation function, the variance-time plot, and the rescaled adjusted range statistic (R/S).

Figure 3 presents the results from the tests for network traffic self-similarity. Figure 3(a) shows the time series under consideration. Figure 3(b) shows the autocorrelation function for this time series. The hyperbolic decay is indicative of self-similarity, though the length of the time series (10,800 data points) is somewhat short to be definitively sure. Figure 3(c) shows a variance-time plot for this time series. The points plotted in the graph show a straight line behaviour with a slope significantly flatter than -1 (the solid line in the graph). This graph suggests a slowly-decaying variance for the aggregated time series,



(a) Time Series Plot (b) Autocorrelation Function (c) Variance-Time Plot (d) R/S Pox Plot

Figure 3. Evidence of Self-Similar Request Arrival Process for Empirical Web Proxy Workload



(a) PDF (b) CDF (c) LLCD

Figure 4. Characteristics of the Request Arrival Process for Empirical Web Proxy Workload

another indication of self-similarity. Finally, Figure 3(d) shows an R/S pox plot for this data set. The slope of this scatter plot can be used to estimate the Hurst parameter H characterizing the degree of self-similarity in this data set. The R/S plot provides a Hurst parameter estimate of $H = 0.74$, again suggesting that the arrival count process is self-similar.

Further detail on the traffic arrival process is provided in Figure 4. Figure 4(a) shows the marginal distribution (i.e., frequency histogram, or probability density function, PDF) of the traffic arrival process from Figure 3(a). The average arrival rate is 20 requests per second, but there is a significant skew to the distribution. Figure 4(b) shows the cumulative distribution function for the arrival process, while Figure 4(c) shows a log-log complementary distribution (LLCD) plot that illustrates the tail behaviour of the distribution. Surprisingly, the downward curvature of Figure 4(c) suggests that the marginal distribution for the count arrival process is *not* heavy-tailed, though the arrival process does appear to be self-similar.

This workload serves as the input to a trace-driven simulation study of cache filter effects in the next section.

UNDERSTANDING FILTER EFFECTS

Experimental Methodology

We use a trace-driven simulation approach for our experimental methodology. The Web workload (a timestamped series of Web document requests) is provided as input to a simple Web proxy cache simulator. The network topology modeled is similar to that shown in Figure 1. The simulator allows configuration of the Web proxy cache size and the cache replacement policy (i.e., which document(s) to remove when the cache is full). The simulator generates as output the cache hit ratio for the experiment, and a timestamped series indicating the requests that produce cache misses. The latter output is called the *filtered request stream*, and is used in our subsequent traffic analyses.

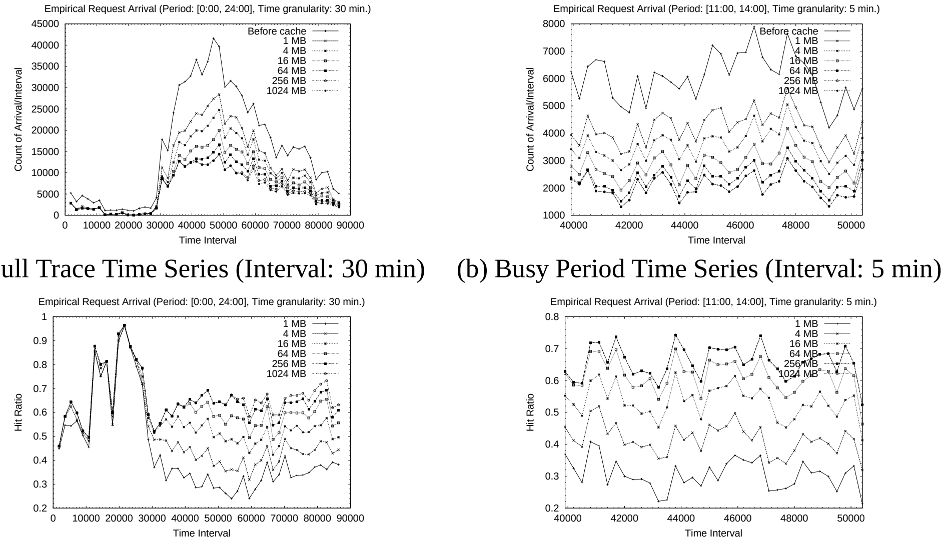
Overview of Cache Filtering Effects

The first simulation experiment illustrates the general impacts of the proxy cache on the Web workload. These cache filter effects are shown in Figure 5.

Figure 5(a) shows the request arrival count time series for the filtered and unfiltered request streams, as a function of time of day. For clarity of presentation, this plot shows the entire workload trace, with request counts sampled over 30 minute intervals. The graph clearly shows that the presence of the Web cache reduces both the peak and the mean rate of the request arrival process. As expected, the larger the cache size is, the more pronounced the filter effect is.

Figure 5(b) shows the same type of plot, but just for the three-hour busy period of the trace (11am to 2pm). In this plot, the request counts are computed over 5 minute intervals. Again, there is a consistent reduction in the mean and peak request rate as the cache size is increased.

Figure 5(c) shows the average document hit ratio in the cache, as a function of time of day, for different cache sizes. Other than the erratic behaviour in the early morning hours (say, 2am to 7am) when few clients are using the cache, the cache hit ratio is relatively stable throughout the day, reflecting “steady state” cache performance for the work-



(a) Full Trace Time Series (Interval: 30 min) (b) Busy Period Time Series (Interval: 5 min)

(c) Full Trace Hit Ratio (Interval: 30 min) (d) Busy Period Hit Ratio (Interval: 5 min)

Figure 5. Illustration of the Proxy Cache Filter Effects on the Empirical Web Proxy Workload

load considered. Note that these results are plotted using the average cache hit ratio over each 30 minute interval of the trace. The cache is initially empty at midnight, and proceeds to fill throughout the day, invoking the cache replacement policy as needed to manage the contents of the cache. Overall, the cache hit ratio tends to increase as the cache size is increased (as expected).

Figure 5(d) shows the corresponding cache hit ratio results for the busy portion of the day from 11am to 2pm. In this plot, the cache hit ratios are computed over 5 minute intervals, with the cache initially empty at midnight. The cache hit ratio clearly increases with the cache size, and is relatively stable throughout the busy period. These observations suggest that the filtered request arrival process for the busy period is likely a stationary process.

Self-Similarity

The purpose of this analysis is to see if the filtered request stream after the Web proxy cache still has the same self-similar properties of the input request stream. As an example, we consider the simple case with a Least-Frequently-Used (LFU) cache replacement policy, with a cache size of 16 MB. We use the same statistical analysis techniques in Figure 3. The results in Figure 6 show that the self-similar characteristics remain present in the filtered request stream. The Hurst parameter is estimated as $H = 0.71$. Our investigations suggest that the self-similar property of the arrival count process is not altered by the presence of the Web proxy cache.

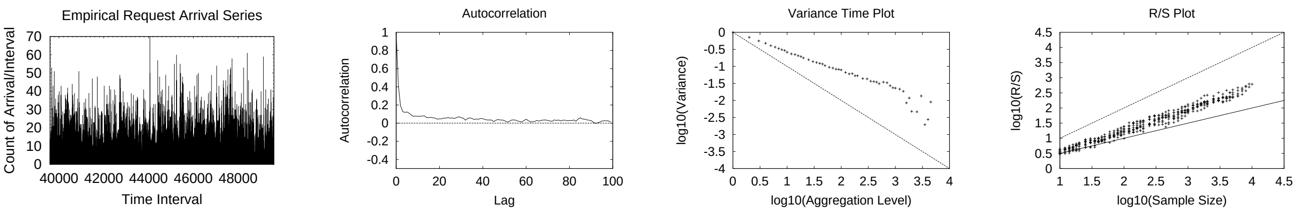
Effect of Cache Size

The next experiment studies the arrival count process for the filtered request stream, as a function of cache size. As was done with the empirical workload trace, we focus on the 11am-2pm busy period of the trace, using arrival counts per one second interval.

Table 2 summarizes these simulation results. Clearly, the presence of the cache reduces both the mean and the standard deviation of the arrival count process after the cache, though the impact on the mean is more pronounced. The larger the cache, the greater this filtering effect. The corresponding cache hit ratios for different cache sizes are also shown in Table 2.

The characteristics of the filtered arrival process are shown in Figure 7. Figure 7(a) shows the marginal distribution (i.e., PDF) of the filtered request streams, for different cache sizes. For ease of reference, the unfiltered request stream is shown using a cache size of 0 MB. The corresponding cumulative distribution functions (CDF) for the arrival count processes are shown in Figure 7(b), and the LLCD plots in Figure 7(c). In general, the filter effect of the cache increases with cache size.

In Figure 7(a), the filter effect of the cache manifests itself in several ways. First, the main “hump” of the marginal distribution moves to the left, reflecting the decrease in the mean arrival rate. Second, the height of the distribution at or near the origin tends to increase, since a large cache produces many one-second intervals with few (or even zero) arriving requests. Third, the distribution tends to decay more quickly (i.e., it has a lighter tail), reflecting the lower variance in the resulting arrival process. The latter two effects together tend to produce a taller and narrower marginal distribution, again reflecting lower variance in the filtered arrival process.

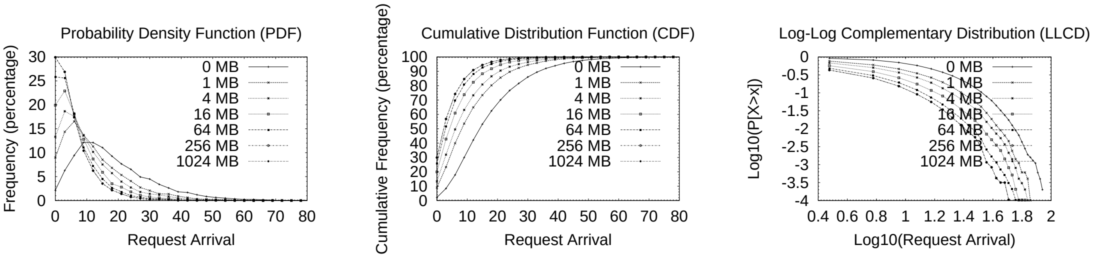


(a) Time Series Plot (b) Autocorrelation Function (c) Variance-Time Plot (d) R/S Pox Plot

Figure 6. Evidence of Self-Similar Request Arrival Process for Filtered Web Proxy Workload

Table 2. Simulation Results for Different Cache Sizes (Empirical Workload, LFU Policy)

Arrival Count Statistics	Before Cache	Cache Size (MB)					
		1	4	16	64	256	1024
Mean	20.27	14.02	11.78	9.38	7.75	6.88	6.88
StdDev	12.41	10.24	9.22	7.85	6.72	6.09	6.09
Hit Ratio	-	33.5%	42.9%	52.5%	59.1%	63.0%	64.1%

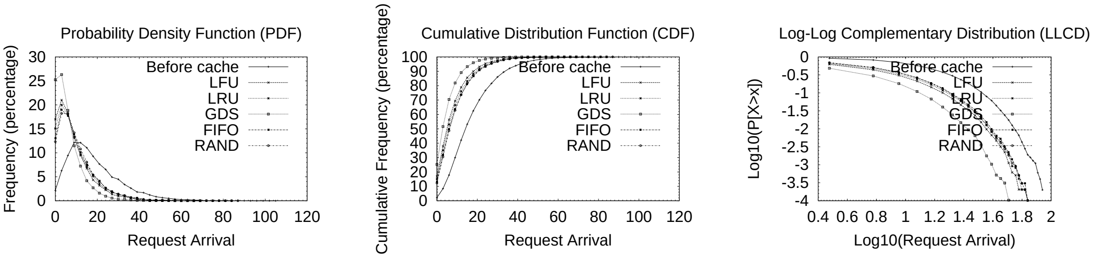


(a) PDF (b) CDF (c) LLCD

Figure 7. Characteristics of the Filtered Arrival Process as a Function of Cache Size (Empirical Workload, LFU Policy)

Table 3. Simulation Results for Different Cache Replacement Policies (Empirical Workload, 8 MB Cache)

Arrival Count Statistics	Before Cache	Cache Replacement Policy				
		RAND	FIFO	LRU	LFU	GDS
Mean	20.27	11.75	11.70	11.05	10.34	7.63
StdDev	12.41	8.82	9.13	8.85	8.39	6.33
Hit Ratio	-	43.0%	43.8%	46.4%	48.6%	60.6%



(a) PDF (b) CDF (c) LLCD

Figure 8. Characteristics of the Filtered Arrival Process as a Function of Cache Replacement Policy (Empirical Workload, 8 MB Cache)

Effect of Cache Replacement Policy

The next experiment looks at the sensitivity of the cache filter effect to the cache replacement policy used. The cache replacement policy determines which document(s) to remove from the cache when more space is needed to store an incoming document. Different cache replacement criteria have been proposed in the literature, such as recency-based, frequency-based, size-based, and so on. We consider five example policies in our study, namely Random replacement (RAND), First-In-First-Out (FIFO), LRU (Least-Recently-Used), LFU (Least-Frequently-Used), and GDS (Greedy-Dual Size). Further details on these policies can be found in the literature [5, 6, 15].

Figure 8 shows the impact of the selected cache replacement policies on the workload characteristics. Examining the plots shows that the filter effects of the policies of LFU and LRU are similar. The GDS policy, however, has the most pronounced impact on the request arrival count process. This difference is due to its higher cache hit ratio. The GDS policy tries to keep small documents in the cache, by associating a weight $H = \frac{1}{s}$ with each document, where s is the size of the document in bytes.

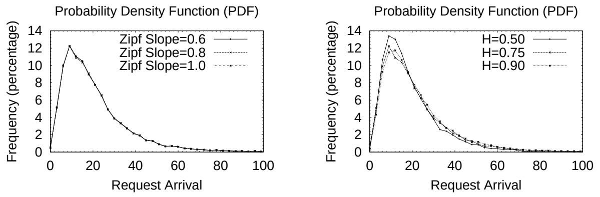
Table 3 summarizes the statistical characteristics of the filtered request arrival process. As expected, the document hit ratio for the GDS policy (60.6%) is higher than for the other policies. FIFO and RAND have less of a filtering effect on the workload, since their “zero knowledge” approach produces a lower cache hit ratio.

Effect of Web Workload Characteristics

To increase the scope of our study, we supplement the foregoing empirical trace with three synthetically-generated Web proxy workload traces. Each trace is generated using the ProWGen (Proxy Workload Generation) tool developed by Busari and Williamson [5].

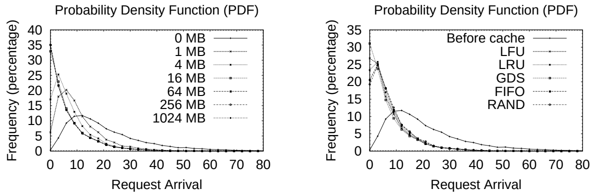
Table 4 summarizes the statistical characteristics of the synthetic traces used. Each trace has approximately 220,000 requests, similar to the busy period of the empirical workload. By design, the three synthetic workloads differ in the slope for the Zipf-like document popularity distribution. Earlier work has shown that the Zipf slope has a significant influence on the cache hit ratio [4, 5, 13]. Trace B has a Zipf slope of 0.8, which closely matches that of the empirical trace used. Trace A has a significantly flatter Zipf slope (0.6), which implies a lower cache hit ratio is expected for this trace. Finally, Trace C has a steeper Zipf slope, meaning that this trace will produce higher cache hit ratios, and thus a more pronounced cache filtering effect.

For each of these three traces, three different traffic arrival processes were generated: a short-range dependent arrival process with $H = 0.5$, a self-similar process with $H = 0.75$ (similar to the empirical workload), and a self-similar process with Hurst parameter $H = 0.9$. Note that the generation of the traffic arrival process (i.e., the timestamps on the Web document requests) is independent of the techniques used to generate the Web document requests (i.e., ProWGen’s file popularity and temporal locality models). Thus the three synthetic traces can be combined with the three traffic arrival models to produce nine synthetic traces with a wide range of workload characteristics. These nine traces are used in the simulations to assess the robustness of our observations about the filtered Web proxy workload characteristics.



(a) $H = 0.75$ (b) $ZipfSlope = 0.8$

Figure 9. Arrival Count PDFs for Synthetic Workloads



(a) Cache Size (b) Replacement Policy

Figure 10. Characteristics of Arrival Count Distribution for Filtered Request Stream (Synthetic Workload, $H = 0.9$, $Z = 0.8$)

Figure 9 provides a graphical validation of the synthetic Web proxy workloads. Figure 9(a) shows that for a given Hurst value H, the distribution of the arrival count process is the same when the Web document popularity distribution in the request stream is changed. Figure 9(b) shows that for a given Zipf slope for the document popularity distribution, the arrival count process varies for $H = 0.5$, $H = 0.75$, and $H = 0.9$. The $H = 0.5$ workload represents a short-range-dependent process. The $H = 0.75$ workload with a Zipf slope of 0.8 is similar in structure to the empirical workload studied. The $H = 0.9$ workload has the highest degree of self-similarity, and thus the longest tail to the arrival count distribution.

Table 5 and Table 6 summarize the simulation results for one representative example of the synthetic workloads ($H = 0.9$, Zipf slope 0.8). Table 5 shows the characteristics of the filtered request stream, as a function of cache size, for an LFU replacement policy. Table 6 shows the characteristics of the filtered request stream, as a function of cache replacement policy, for a fixed-size 8 MB cache. The characteristics of the filtered request stream are qualitatively similar to those for the empirical trace.

Figure 10 shows the graphical characteristics of the fil-

Table 4. Characteristics of Synthetic Web Proxy Workloads

Item	Trace A	Trace B	Trace C
Trace Duration	3 hours	3 hours	3 hours
Total Requests	225,042	218,845	225,697
Total Transferred Bytes (Mbytes)	2,144	1,871	1,654
Mean Transfer Size (bytes)	9,991	8,967	7,683
Median Transfer Size (bytes)	3,552	3,474	3,285
Total Documents	70,254	70,870	70,951
Unique Documents (% of requests)	31.2%	32.4%	31.4%
Total Bytes of Documents (Mbytes)	793	798	799
Smallest Document Size (bytes)	34	34	34
Largest Document Size (bytes)	12,382,599	12,382,599	12,382,599
Mean Document Size (bytes)	11,830	11,808	11,806
Median Document Size (bytes)	3,817	3,816	3,815
One-timer Documents	48,942	49,558	49,638
One-timers (% of documents)	69.7%	69.9%	70.0%
Zipf Slope	-0.6	-0.8	-1.0

Table 5. Simulation Results for Different Cache Sizes (Synthetic Workload, $H = 0.9$, $Z = 0.8$, LFU Policy)

Arrival Count Statistics	Before Cache	Cache Size (MB)					
		1	4	16	64	256	1024
Mean	23.60	12.67	9.75	7.83	7.65	7.64	7.64
StdDev	16.80	9.01	8.14	8.28	8.32	8.32	8.32
Hit Ratio	-	46.3%	58.7%	66.8%	67.6%	67.6%	67.6%

Table 6. Simulation Results for Different Replacement Policies (Synthetic Workload, $H = 0.9$, $Z = 0.8$, 8 MB Cache)

Arrival Count Statistics	Before Cache	Cache Replacement Policy				
		RAND	FIFO	LRU	LFU	GDS
Mean	23.60	9.62	9.35	8.84	8.48	8.03
StdDev	16.80	8.36	8.24	8.06	8.12	8.23
Hit Ratio	-	59.2%	60.4%	62.5%	64.1%	66.0%

tered request arrival process. For space reasons, only the results for $H = 0.9$ and a Zipf slope of 0.8 are shown. The qualitative behaviour in Figure 10(a) and (b) is consistent with that observed for the empirical Web proxy trace. Larger cache sizes produce a leftward shift of the distribution, and an increase in its peak value at the origin. This behaviour is consistent for all Hurst parameter values and Zipf slope values considered in our experiments, though the leftward shift of the distribution is (as expected) more pronounced at smaller cache sizes as the Zipf slope increases.

MODELING CACHE FILTER EFFECTS

This section discusses a parameterizable mathematical model for characterizing the request arrival count distribution both before and after a Web proxy cache. Prior experience with network traffic modeling provides intuition that a Gamma distribution may be suitable for modeling the (filtered or unfiltered) arrival count distribution for Web workloads, since the shape of the distributions in Figure 7 and elsewhere are reminiscent of the Gamma distribution. The next section provides some background on the Gamma distribution, while the section after that validates the Gamma distribution model on the empirical workload.

Background

The general formula for the Gamma distribution probability density function (PDF) is:

$$f(x) = \frac{(\frac{x-\mu}{\beta})^{\gamma-1} e^{(-\frac{x-\mu}{\beta})}}{\beta\ \Gamma(\gamma)} \qquad x \geq \mu; \quad \gamma, \beta > 0 \tag{1}$$

where γ is the *shape* parameter, μ is the *location* parameter, β is the *scale* parameter, and Γ is the Gamma function

$$\Gamma(a) = \int_{0}^{\infty} t^{a-1} e^{-t} dt \tag{2}$$

The case where $\mu = 0$ and $\beta = 1$ is called the standard Gamma distribution. The equation for the standard Gamma distribution reduces to:

$$f(x) = \frac{x^{\gamma-1} e^{-x}}{\Gamma(\gamma)} \qquad x \geq 0; \quad \gamma > 0 \tag{3}$$

Figure 11 shows examples of the probability density function for the standard Gamma distribution with different choices of shape parameter γ. As γ decreases, the center of gravity of the distribution moves to the left, the peak value of the curve increases, and the tail of the curve decreases more quickly. If $\gamma \leq 1$, then the distribution is monotonically decreasing. These behaviours are similar to those for the empirical and synthetic request arrival count processes in our study, suggesting the suitability of the Gamma distribution for our traffic modeling purposes.

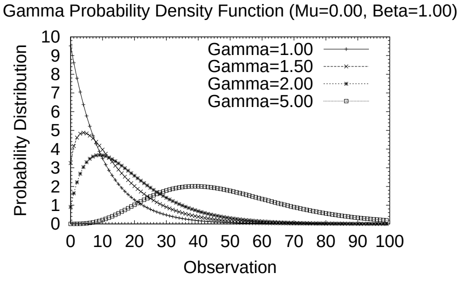


Figure 11. Gamma Probability Density Function

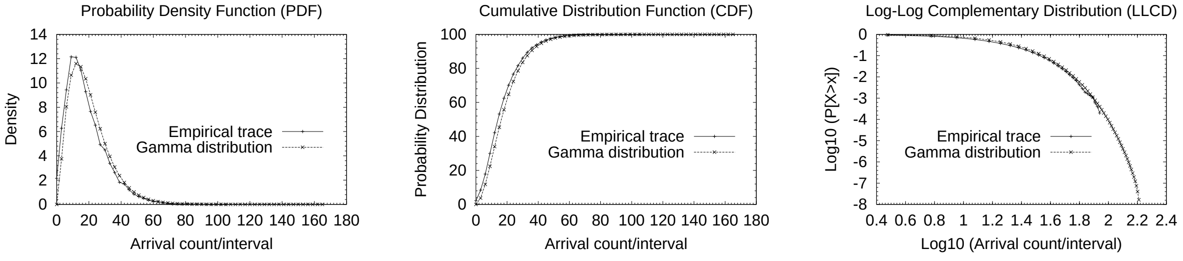
Modeling the Arrival Count Distribution

Figure 12 illustrates the characteristics of the request arrival count distribution for the empirical workload, along with a Gamma model fit to the distribution. The parameters γ and β of the Gamma distribution ($\gamma = 2.67$, $\beta = 7.60$) were determined using *maximum likelihood estimates*. Figure 12 shows that the Gamma distribution provides a good visual fit of the distribution, for both the body and the tail of the distribution.

Figure 13 shows similar plots for the filtered request arrival process after the cache. This particular plot illustrates the results for a 4 MB LFU cache at the Web proxy. The Gamma distribution with $\gamma = 1.63$ and $\beta = 7.22$ provides a good visual fit to the traffic arrival distribution, in both the body and the tail. Again, the parameters of the Gamma distribution were determined using maximum likelihood estimates.

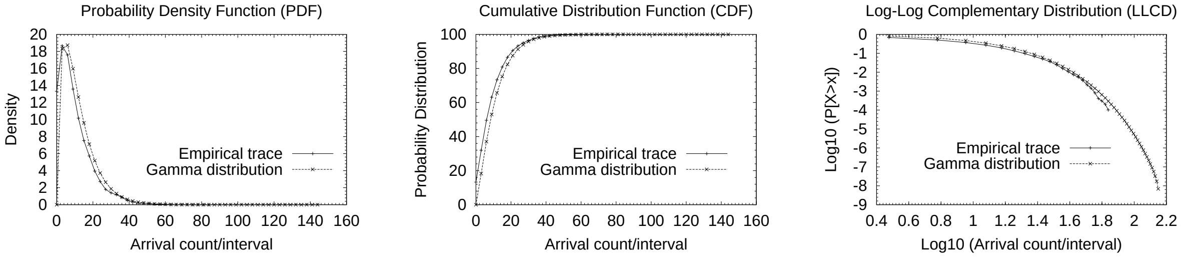
Table 7 summarizes the maximum likelihood estimates (MLE) of the γ and β parameters, for different cache sizes. As the cache size (and thus the cache hit ratio) increases, the MLE of the γ (shape) parameter of the distribution decreases, reflecting the general leftward movement of the body of the distribution illustrated in Figure 7. In all of our experiments with empirical and synthetic traces, the MLE of the γ parameter is a monotonically-decreasing function of cache size (since the cache hit ratio monotonically increases).

Table 7 shows that the MLE of the β parameter also decreases with an increase in cache size, reflecting the taller vertical height of the distribution as the cache hit ratio increases. However, the decrease in β is not as pronounced as the decrease in γ. Furthermore, this monotonically decreasing behaviour for β was not observed for all workload traces studied. Several of the synthetic Web proxy workloads had non-monotonic relationships for β. In particular, β tends to increase once $\gamma \leq 1$. Clearly, the γ and β parameters depend on the characteristics of the input Web proxy workload, as well as the cache parameters (since these together influence the cache hit ratio). We are currently trying to quantify and understand these mathematical relationships, since they are crucial to modeling Web traffic in multi-level Web proxy caching architectures.



(a) PDF (b) CDF (c) LLCD

Figure 12. Gamma Distribution Model for Empirical Request Arrival Count Distribution ($\gamma = 2.67$, $\beta = 7.60$)



(a) PDF (b) CDF (c) LLCD

Figure 13. Gamma Distribution Model for Filtered Arrival Count Distribution (Empirical Workload, $\gamma = 1.63$, $\beta = 7.22$

Table 7. Maximum Likelihood Estimates for Gamma Distribution Parameters (Empirical Workload, LFU Policy)

Estimated Parameters	Before Cache	Cache Size (MB)					
		1	4	16	64	256	1024
$\hat{\gamma}$	2.67	1.87	1.63	1.43	1.33	1.28	1.28
$\hat{\beta}$	7.60	7.48	7.22	6.57	5.83	5.39	5.39

SUMMARY AND CONCLUSIONS

This paper used trace-driven simulation to study the structural characteristics of Web workloads both before and after a Web proxy cache. The paper focuses on time-domain analysis of cache filter effects; that is, it focuses on the statistical characteristics of the request arrival count process as transformed by the cache.

Our simulation results demonstrate that the cache reduces both the mean arrival rate and the peak arrival rate, but has relatively little impact on the variability and the self-similarity of the request arrival process. We find that the Gamma distribution provides a flexible and robust model for characterizing the request arrival process, though the parameters for the Gamma distribution are highly dependent upon cache size and Web workload characteristics.

Our current work is proceeding along three fronts. First, we are analyzing more (and longer) Web proxy access logs, to determine the generality of our modeling results, and to make a more rigourous assessment of network traffic self-similarity. Second, we are attempting to derive the mathematical relationships between cache size, Zipf slope, Web workload characteristics, and the Gamma distribution parameters of our model. Better understanding of these relationships will lead to insights about traffic aggregation and superposition in multi-cache Web proxy caching architectures. Third, we are striving to apply our characterization and modeling techniques to the downstream (response) traffic direction as well, rather than just to the upstream request arrival process. This effort will allow us to quantify the true benefits of Web proxy caching.

ACKNOWLEDGMENTS

Financial support for this research was provided by iCORE (Informatics Circle of Research Excellence) and by the Natural Sciences and Engineering Research Council of Canada. The authors are grateful to Greg Oster at the University of Saskatchewan for making traces from the U of S Web proxy available.

References

[1] V. Almeida, M. Cesario, R. Fonseca, W. Meira Jr., and C. Murta, “Analysing the Behavior of a Proxy Server in Light of Regional and Cultural Issues”, *Proceedings of the Third International WWW Caching Workshop*, Manchester, England, June 1998.

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

[3] 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.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web Caching and Zipf-like Distributions: Evidence and Implications”, *Proceedings of IEEE INFOCOM*, New York, NY, pp. 126-134, March 1999.

[5] M. Busari and C. Williamson, “On the Sensitivity of Web Proxy Cache Performance to Workload Characteristics”, *Proceedings of IEEE INFOCOM*, pp. 1225-1234, Anchorage, AL, April 2001.

[6] M. Busari and C. Williamson, “Simulation Evaluation of a Heterogeneous Web Proxy Caching Hierarchy”, *Proceedings of MASCOTS*, pp. 379-388, Cincinnati, OH, August 2001.

[7] H. Che, Z. Wang, and Y. Tung, “Analysis and Design of Hierarchical Web Caching Systems”, *Proceedings of IEEE INFOCOM*, pp. 1416-1424, Anchorage, AL, April 2001.

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

[9] R. Doyle, J. Chase, S. Gadde, and A. Vahdat, “The Trickle-Down Effect: Web Caching and Server Request Distribution”, *Proceedings of the Web Caching and Content Delivery Workshop*, Boston, MA, June 2001.

[10] M. Franklin, M. Carey, and M. Livny, “Global Memory Management for Client-Server DBMS Architectures”, *Proceedings of the 19th International Conference on Very Large Databases* (VLDB), August 1992.

[11] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, “On the Self-Similar Nature of Ethernet Traffic (Extended Version)”, *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, pp. 1-15, February 1994.

[12] A. Mahanti, C. Williamson, and D. Eager, “Traffic Analysis of a Web Proxy Caching Hierarchy”, *IEEE Network*, Vol. 14, No. 3, pp. 16-23, May/June 2000.

[13] C. Roadknight, I. Marshall, and D. Vearer, “File Popularity Characterization”, *Proceedings of the Second Workshop on Internet Server Performance (WISP’99)*, Atlanta, Georgia, May 1999.

[14] D. Weikle, S. McKee, and W. Wulf, “Caches as Filters: A New Approach to Cache Analysis”, *Proceedings of MASCOTS*, Montreal, PQ, pp. 2-12, July 1998.

[15] C. Williamson, “On Filter Effects in Web Caching Hierarchies”, *ACM Transactions on Internet Technology*, Vol. 2, No. 1, pp. 47-77, February 2002.

[16] D. Willick, D. Eager, and R. Bunt, “Disk Cache Replacement Policies for Network Fileservers”, *Proceedings of ICDCS*, Pittsburgh, PA, 1993.