In Proceedings of PDIS'96: The IEEE Conference on Parallel and Distributed Information Systems, Miami Beach, FL, Dec 1996.

Characterizing Reference Locality in the WWW*

Virgílio Almeida[†§] (virgilio@bu.edu)
Azer Bestavros[†] (best@bu.edu)
Mark Crovella[†] (crovella@bu.edu)
Adriana de Oliveira[‡] (dri@dcc.ufmg.br)

Abstract

In this paper we propose models for both temporal and spatial locality of reference in streams of requests arriving at Web servers. We show that simple models based on document popularity alone are insufficient for capturing either temporal or spatial locality. Instead, we rely on an equivalent, but numerical, representation of a reference stream: a stack distance trace. We show that temporal locality can be characterized by the marginal distribution of the stack distance trace, and we propose models for typical distributions and compare their cache performance to our traces. We also show that spatial locality in a reference stream can be characterized using the notion of self-similarity. Self-similarity describes long-range correlations in the dataset, which is a property that previous researchers have found hard to incorporate into synthetic reference strings. We show that stack distance strings appear to be stongly self-similar, and we provide measurements of the degree of self-similarity in our traces. Finally, we discuss methods for generating synthetic Web traces that exhibit the properties of temporal and spatial locality that we measured in our data.

1 Introduction

The principle of locality of reference has very important consequences for computer systems design. Reference streams exhibiting *temporal* locality can benefit from caching; and reference streams exhibiting *spatial* locality can benefit from prefetching. The application of these principles (*e.g.*, in memory systems) is well understood [10, 23].

In order to apply these principles to the design of World Wide Web (Web) caching and prefetching systems, it's important to characterize the degree of locality present in typical Web reference streams. For Web accesses, temporal locality refers to the property that an item frequently accessed in the past is likely to be accessed in the future, whereas spatial locality refers to the property that items *neighboring* an item frequently accessed in the past are likely to be accessed in the future. For memory systems, such *neighborhoods* are easily defined in terms of address space proximity, giving rise to prefetching entities such as *cache lines* and *memory pages*. For the Web, we believe that the identification of similar prefetching entities is possible if spatial locality is characterized properly.

Previous research in the locality properties of symbolic reference streams have shown the benefits of transforming the reference stream into an equivalent, but numerical, representation: a *stack distance* stream [19]. This transformation preserves all of the information of the original trace except for the specific reference names, and so it is invertible, allowing reconstruction of the original trace (albeit with synthetic names). Two properties of the stack distance stream can be used to measure the notions of temporal and spatial locality: *marginal distribution,* and *correlation structure* [19, 22]. The marginal distribution measures the likelihood that two references to the same object are separated by some number of intervening references in the stream. The correlation structure of the stream measures the likelihood that we can predict future references based on the stream of past references.

In this paper we use quantitative methods based on stack distance to measure both the temporal and spatial locality properties for Web reference streams. We characterize reference streams collected at four Web servers, propose a model for marginal distribution of stack distance (and thus temporal locality) and show typical ranges of the relevant parameters. We also propose a model for spatial locality of our reference streams based on the correlation structure of stack distances. Using these measurements we show how to generate synthetic reference streams of arbitrary length whose performance properties mimic those of our empirically measured data.

A novel feature of our approach is the use of statistical *self-similarity* to capture the correlation structure of the reference stream. That is, we show evidence that Web stack distance streams are statistically self-similar and that this property can be used to explain aspects of the reference stream that are the result of spatial locality. The presence of self-similarity in the reference stream means that correlations between object references can occur at widely varying timescales (*long-range depen-*

---

[*] This work has been partially supported by NSF (grants CCR-9308344 and CCR-9501822) and by CNPq-Brazil.

[†] Computer Science Department, Boston University, Boston, MA 02215, USA.

[‡] Depto. de Ciencia da Computacão, Universidade Federal de Minas Gerais, Belo Horizonte, MG 30161, Brazil.

[§] On sabbatical at Boston University.

dence). A timeseries with long-range dependence appears much more "bursty" than series with only short-range or no dependence — which agrees with generally observed characterizations of reference locality. Long-range dependence in the stack distance series is also related to previously noted fractal properties of cache miss patterns [26], and we interpret prior work in fractal reference patterns in terms of our observations.

This method of measuring spatial locality also provides an interpretation for the notion of spatial locality when reference streams are composed of symbols rather than numbers. We consider object A to be close to object B if there is a high likelihood that a reference to A will immediately follow a reference to B. That is, the "distance" between objects is the inverse of the transition probability of object references considered as a first-order Markov chain. While this definition does not correspond to our usual notions of a norm (it is not symmetric), it does capture the important property of spatial locality for the purposes of prefetching: the likelihood that prefetching will be successful.

Our methods focus on the locality properties of reference streams without regard to document size. Thus in our simulations we consider caches to be sized to hold a fixed number of documents rather than a fixed amount of data. This assumption stems from the fact that we are interested in locality, which is a property of reference streams independent of any cache size.[2]

In the remainder of this section we describe the data on which our measurements and trace simulations are based. In section 2, we characterize the popularity of Web documents and show that a cache reference model based on popularity is not expressive enough to capture the temporal and spatial locality of reference properties exhibited in actual Web traces. Evidence of these locality of reference properties is presented in section 3. In sections 4 and 5, we characterize these properties and present statistical models that capture these characteristics. In section 6, we examine related work. We conclude in section 7 with a summary and directions for future work.

Data Collection: The Web is a large-scale distributed information system based on a client-server architecture. Thus, the workload viewed from a server standpoint consists of a number of requests originated at many different clients. In order to analyze reference locality, we examined the access logs for different Web servers, namely: the NCSA Web server located at the National Center for Supercomputing Applications (NCSA), the SDSC Web server at the San Diego Supercomputer Center (SDSC), the EPA Web server located at Research Triangle Park, NC and the Web server at the Computer Science Department at Boston University (BU). In the case of the NCSA, the data refers only to one server (Costello). The SDSC and EPA logs are available at the Internet Traffic Archives [14], the NCSA log was obtained after contacting the staff at the site and the BU logs were collected at the departmental Web server.

The logs have one line of information per request processed by the server. Each line contains the name of the host making the request, the timestamp the request was made, the filename of the requested object and size in bytes of the reply. Table 1 summarizes the statistics about the logs of the four Web servers.

Log	**NCSA**	**SDSC**	**EPA**	**BU**
Duration	1 day	1 day	1 day	2 weeks
Start Date	Dec 19	Aug 22	Aug 29	Oct 08
Total requests	46,955	28,338	47,748	80,518
Unique requests	4,851	1,267	6,518	4,471

Table 1: Summary of Access Log Data

2 Characterizing Document Popularity

The highly uneven popularity of various Web documents is a well-documented phenomenon that has been exploited in previous caching, replication and dissemination studies. In these studies, it was shown that "Popular files are very popular" [4, 13]. In [9], Cunha, Bestavros, and Crovella characterized the popularity of *Web documents requested by clients* and confirmed the applicability of Zipf's law [27, discussed in [17]] to Web documents. Zipf's law was originally applied to the relationship between a word's popularity in terms of rank and its frequency of use. It states that if one ranks the popularity of words used in a given text[3] (denoted by ρ) by their frequency of use (denoted by P) then

$$P \sim 1/\rho.$$

Note that this distribution is a parameterless hyperbolic distribution. *i.e.,* ρ is raised to exactly -1, so that the n^{th} most popular document is exactly twice as likely to be accessed as the $2n^{\text{th}}$ most popular document.

Our data shows that Zipf's law applies quite strongly to *Web documents serviced by Web servers.* This is demonstrated in Figure 1 for all 2135 documents accessed in the BU trace. The figure shows a log-log plot of

---

[2] Since we have shown in previous work [9] that document size is correlated with expected reference frequency, this assumption is somewhat limiting for direct use in cache performance analysis. For that reason we are currently extending this work to include document sizes in our analysis. Nonetheless we believe that the current methods provide a simple and effective way to capture both the spatial and temporal locality of Web reference streams.

[3] Zipf's law has subsequently been applied to other examples of popularity in the social sciences.

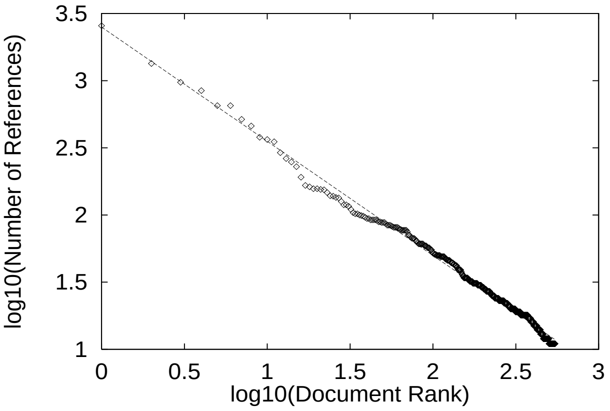

Figure 1: Zipf's Law Applied to Web Documents

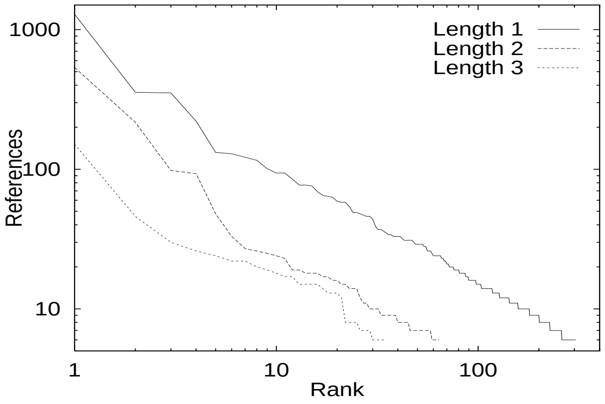

Figure 2: Zipf's Law for Web Document Sequences

the total number of references (Y axis) to each document as a function of the document's rank in overall popularity (X axis). The tightness of the fit to a straight line is strong ($R^2 = 0.99$), as is the slope of the line: -0.85 (shown in the figure). Thus the exponent relating popularity to rank for Web documents is very nearly -1, as predicted by Zipf's law.

Our measurements suggest that Zipf's law applies, not only to single requests for documents, but also to request *strides*. A request stride is a sequence of requests from the *same* client where the time between successive requests in that stride is not larger than a given *stride threshold*. Figure-2 shows a log-log plot of the total number of references (Y axis) to strides of length k (for $k = 1, 2, 3$) as a function of the stride's rank in overall popularity (X axis). The trace used for this plot is the BU trace and the stride threshold is set at 10 seconds. Again, the tightness of the fit for the family of curves in Figure-2 to straight lines slope close to -1 suggest that Zipf's law does apply to sequences of requests.

Given the above observations, one possible access model [12, 2] is to generate document requests according to the popularity profile of documents. We call this model the *Zipf-based model*. To test the validity of this model we measured the miss rate at the server when requests are generated synthetically, to satisfy the observed popularity profile of documents at BU, and compared these measurements with the measurements obtained when the actual BU trace is used to drive the simulation. The synthetic trace was generated by applying a random permutation to the actual trace for BU. The result of this comparison is shown in Figure 3.

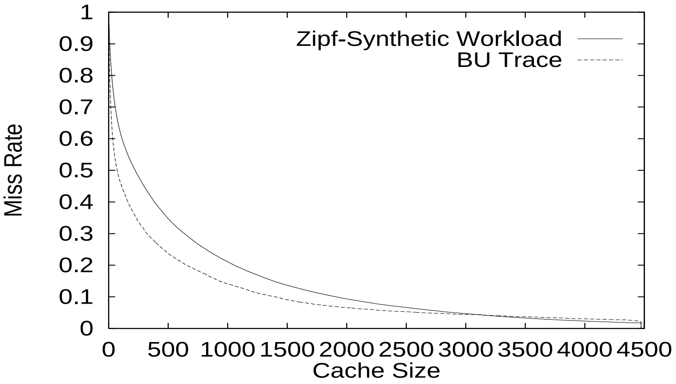

Figure 3: Miss rate for a Zipf-based synthetic workload and for an actual trace

Obviously, using popularity as the basis of a cache model for Web documents does not yield acceptable results. Figure 3 shows that for a cache size of 400 items, the miss rate that results from a Zipf-based cache model is nearly 33% less than the actual miss rate.

The main weakness of a Zipf-based model is that it does not capture the potential locality of reference properties that may be present in actual traces. In particular, while the Zipf-based synthetic trace preserves the general popularity profile of documents, it fails to capture the temporal and spatial locality of reference properties. In the next section we show evidence that these properties indeed exist in Web traces, and in sections 4 and 5, we provide simple, yet expressive statistical models thereof.

3 Evidence of Reference Locality

Temporal Locality: Temporal locality implies that recently accessed documents are more likely to be referenced in the near future. Based on the concepts proposed in [19, 24], we define a stack distance model that captures the temporal locality relationships present in a request stream. Let us consider a reference stream $R_t = r_1, r_2, ..., r_t$, where r_t denotes the name of the object requested at virtual time t. Our unit of time will be one request, so that virtual time t indicates that t requests have already arrived at a server. Let us also define the LRU stack S_t, which is an ordering of all objects of a server by recency of usage. Thus, at virtual time t, the LRU stack is given by:

$$S_t = \{o_1, o_2, \ldots, o_N\} \tag{1}$$

where $o_1, o_2, \ldots, o_N$ are objects of the server and o_1 is the most recently accessed object, o_2 the next most recently referenced, etc. If o_1 is the most recently accessed document, then $r_t = o_1$. Whenever a reference is made to an object, the stack must be updated. Considering that $r_{t+1} = o_i$, then the stack becomes $S_{t+1} = \{o_i, o_1, o_2, ..., o_{i-1}, o_{i+1}, ..., o_N\}$. Suppose now that $S_{t-1} = \{o_1, o_2, ..., o_N\}$ and $r_t = o_i$. Then, we could say that request r_t is at distance i in stack S_{t-1}. Let d_t denote the stack distance of the document referenced at time t. We then have the following relation:

$$\text{if } r_t \;=\; o_i \text{ then } d_t \;=\; i \tag{2}$$

Thus, for any request string $R_t = r_1, r_2, ..., r_t$ there is a corresponding distance string $\delta = d_1, d_2, ..., d_t$. We can then consider that the distance string and the request string are equivalent in terms of reference information. Since the distance string reflects the pattern in which Web users request documents from a server rather than the actual identity of the documents, it will be used as our model for object reference pattern.[4]

The marginal distribution of distance probabilities is an indication of temporal locality because it measures the number of intervening references between two references to the same object. Small stack distances result from frequent references to a document. In other words, the documents referenced most frequently tend to be those close to the top of the stack.

As a measure of the temporal locality present in our traces, we compare the average stack distance found in the BU trace with that of the scrambled BU trace. The results, shown in table 2, indicate that the average stack distance is apparently greater for the scrambled trace, meaning that temporal locality has been decreased.

Spatial Locality: The existence of spatial locality of reference could be established by comparing the total number of unique sequences observed in a trace and the total number of unique sequences that would be found in a random permutation of such a trace. Notice that a random permutation of a trace (henceforth called a *scrambled* trace) *preserves* the popularity profile of the various documents in that trace, but it *destroys* the spatial locality of reference that may exist in such a trace (by uncorrelating the sequence of references). If references are indeed correlated, then one would expect the total number of unique sequences observed in a trace to be much less than the total number of unique sequences observed in a scrambled trace.

BU Trace	Original	Scrambled
Mean Stack Distance	479.798	645.586
Standard Deviation	941.430	968.840

Table 2: Evidence of Temporal Locality

Figure 4 shows the total number of unique sequences (Y axis) of length k (X axis) observed in the first week (the first 32,839 references) of the BU trace, which we call the BU1 server trace. Three cases are shown. The top curve shows the total number of unique k-long sequences observed in a scrambled BU1 trace. The middle curve shows the total number of unique k-long sequences observed in the BU1 trace. Since traces arriving at the server reflect an interleaving of a number of individual client traces, we plot in the bottom curve the total number of unique k-long sequences observed when separately considering individual clients in the BU1 trace.

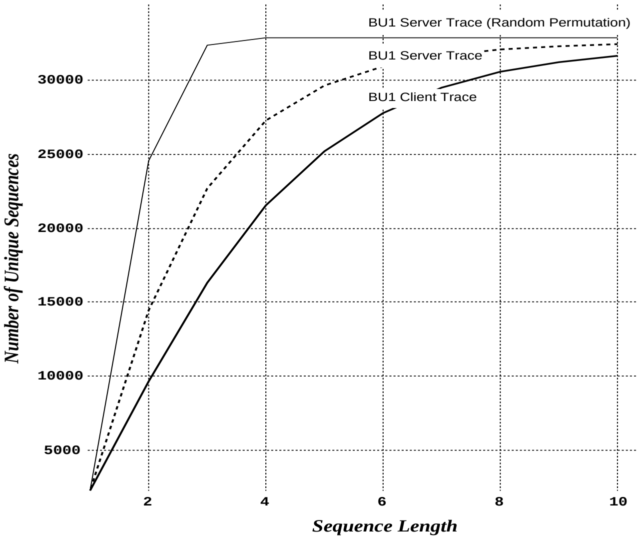

Figure 4: Unique sequences observed in the BU trace

All three curves show an increase in the total number of unique sequences as k increases,[5] with the random-

---

[4] The particular choice of LRU as the stack ordering algorithm does not restrict the applicability of our methods to use with any particular caching strategy; however it does allow direct evaluation of LRU caches as a means of testing the accuracy of our methods, which we will do in Section 4.

[5] The reason that all three curves level-off is related to the limited length of the trace. In particular, it is impossible to observe

ized trace showing the steepest increase and the client trace showing the slowest increase. These results confirm the existence of considerable spatial locality in client access patterns. For example, while up to 32,327 different sequences of length 3 are observed in the scrambled BU1 trace, only 16,114 (more than 50% less) different sequences of length 3 are observed in the BU1 client traces. This spatial locality is “diluted” to some extent (but still very much evident) in the BU1 server trace—22,680 (about 30% less) different sequences of length 3 are observed in the BU1 server trace.

4 Characterizing Temporal Locality

Since it is clear that real traces and scrambled traces differ significantly in the aggregate properties of their stack distance traces, we examine the marginal distributions of stack distances.

As discussed in Section 1, the marginal distribution of stack distance captures the temporal locality present in the trace. That is, if D is a random variable corresponding to stack distance (with distribution function F_D) and $M(C)$ is the miss rate of a cache that can hold C files (using the same replacement algorithm that was used to generate the stack distance trace), then:

$$P[D > C] = 1 - F_D(C) = M(C)$$

Thus, knowledge of the complementary distribution function $1 - F_D$ provides enough information to predict the performance of a cache of any size for the given trace.

To estimate distributions for the four traces we consider, histograms are shown in Figure 5. This figure shows that most of the stack distances are close to 1 (the traces show good temporal locality) and yet there are long tails to the distributions. We considered a number of long-tailed distributions to model this data. In particular we rejected the Pareto distribution ([8]) because although the tails are long, they do not seem to follow a power-law. The data were found to be best fit by a *lognormal* distribution. Data that follows a lognormal distribution has the property that its logarithm is normally distributed. The resulting distribution has a very long tail but is not *heavy-tailed* in the strict sense of following a power-law in the tails.

To illustrate the use of lognormal distributions to model stack distances, Figure 6 shows the empirically measured distributions and the fitted distributions. Each plot shows a histogram of the base 10 logs of the stack distance data; in addition the plot shows the shape of the corresponding fit of a lognormal distribution to the data. While the data sets may appear to vary from normal to some degree, in reality these differences have been magnified by the log-transformation process. In fact the fit to the data is quite close as we will show below.

The parameters used to generate the fitted distributions shown in Figure 6 are given in Table 3. The $\hat{\mu}$ and $\hat{\sigma}$ columns show the mean and standard deviation *of the base 10 log* of the datasets. The table shows that for the traces we studied, the $\hat{\sigma}$ varies only slightly. Most of the variation between the datasets occurs in the $\hat{\mu}$ parameter.

	BU	NCSA	SDSC	EPA
$\hat{\mu}$	1.829	1.730	1.568	2.150
$\hat{\sigma}$	0.947	0.836	0.827	0.921

Table 3: Lognormal Distribution Parameters for Traces

The ability of lognormal distributions to predict actual miss rates in finite caches is shown in Figure 7. As discussed above we plot $P[D > C]$ since it is equivalent to miss rate. In each plot, we show the predicted miss rate using the lognormal stack distance model and the actual miss rate obtained from the data. The agreement between model and data in each case is good, except in the case of the EPA trace, which has very poor locality.

A comparison of the traces in terms of their fitted parameters is shown in Figure 8. This figure shows how temporal locality (measured by miss rate) varies across the traces, and the corresponding parameter values that capture that locality. The figure shows that as the temporal locality of a trace decreases, the mean of lognormal distribution increases (from about 1.6 up to 2.2) and the standard deviation of the lognormal distribution also tends to increase, although to a lesser degree (from 0.83 up to 0.95).

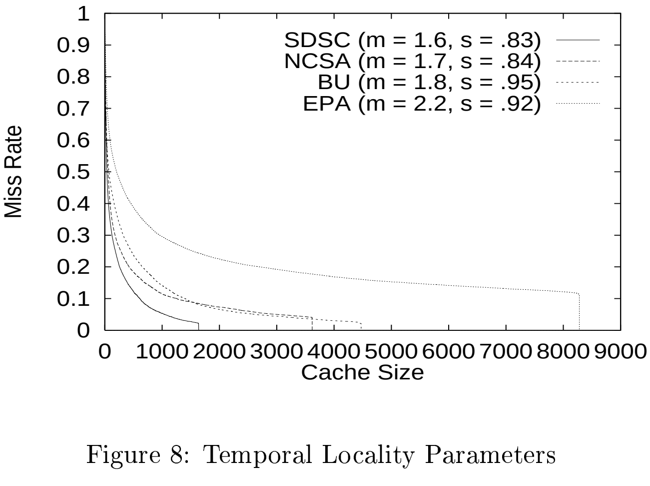

Figure 8: Temporal Locality Parameters

---

any more than $l - k$ unique k-long sequences in a trace of length l. For our experiments $l = 32,839$—hence the observed ceiling.

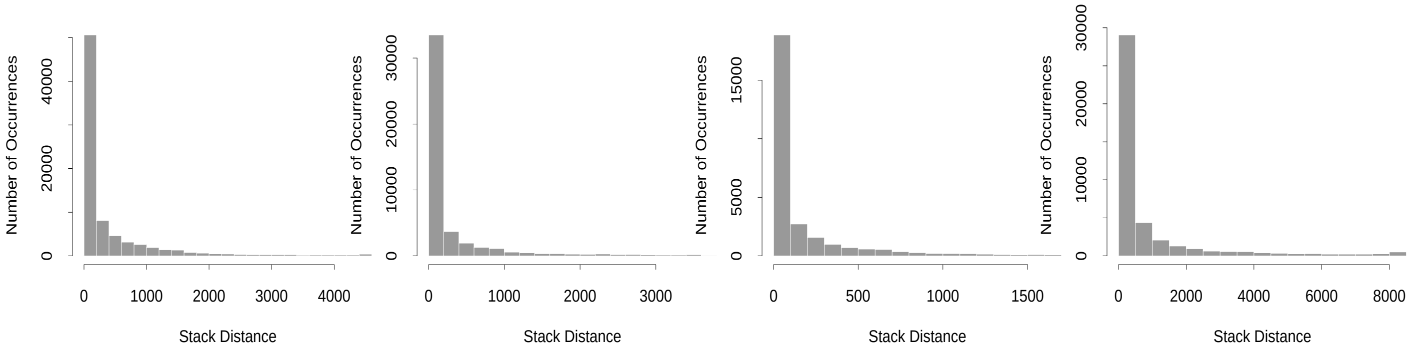

Figure 5: Empirical Stack Distance Distributions (Left to Right: BU, NCSA, SDSC, EPA)

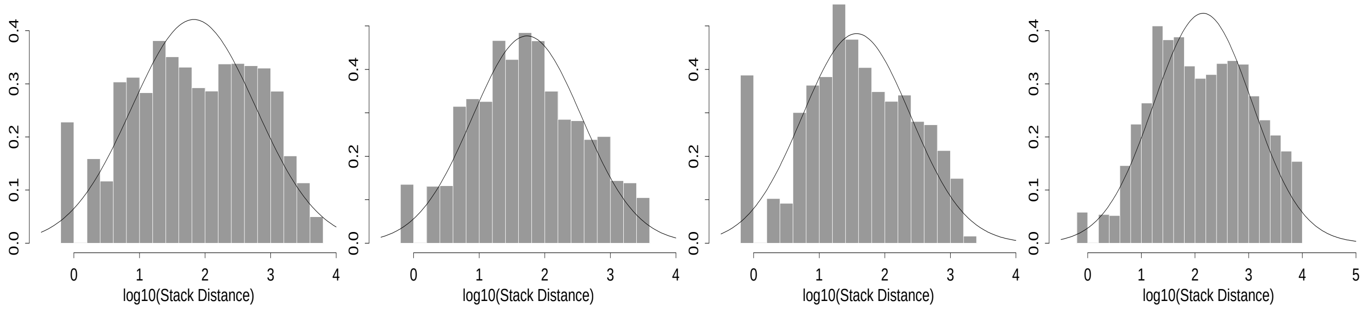

Figure 6: Lognormal Distributions of Stack Distance Traces (Left to Right: BU, NCSA, SDSC, EPA)