Parallel and Distributed Information Systems, Miami Beach, FL

Characterizing Reference Locality in the WWW

Azer Bestavros & *Mark Crovella*
Computer Science Department
Boston University

Virgilio Almeida & *Adriana de Oliveira*
Computer Science Department
Universidade Federal de Minas Gerais

Thursday December 19^{th} 1996

Talk Outline

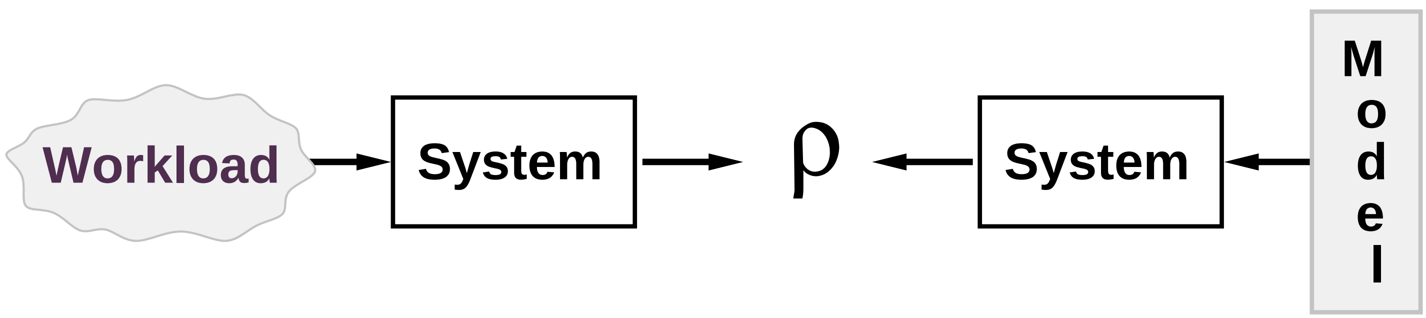
- Introduction, Motivation, and Applications
- Experimental Environment and Data Collection
- Characterizing Web Document Popularity
- Characterizing Web Reference Locality
  - Temporal Locality: Evidence and Model
  - Spatial Locality: Evidence and Model
- Synthetic Web Reference Trace Generation
- Related Work
- Current and Future Work

Introduction

- The characteristics of Web access patterns fall into two categories:
  - Static (*e.g.* popularity profiles)
  - Dynamic (*e.g.* reference locality)

- Characterizing Web access patterns is crucial for performance tuning and evaluation.
  - Client/server caching and prefetching protocols
  - Scheduling and load balancing protocols
  - Networking issues

Evaluating A Workload Model



- A workload model W is a perfect representation of the real workload R if the performance metrics ρ obtained using W and R in the same system are indistiguishable.

Data Collection

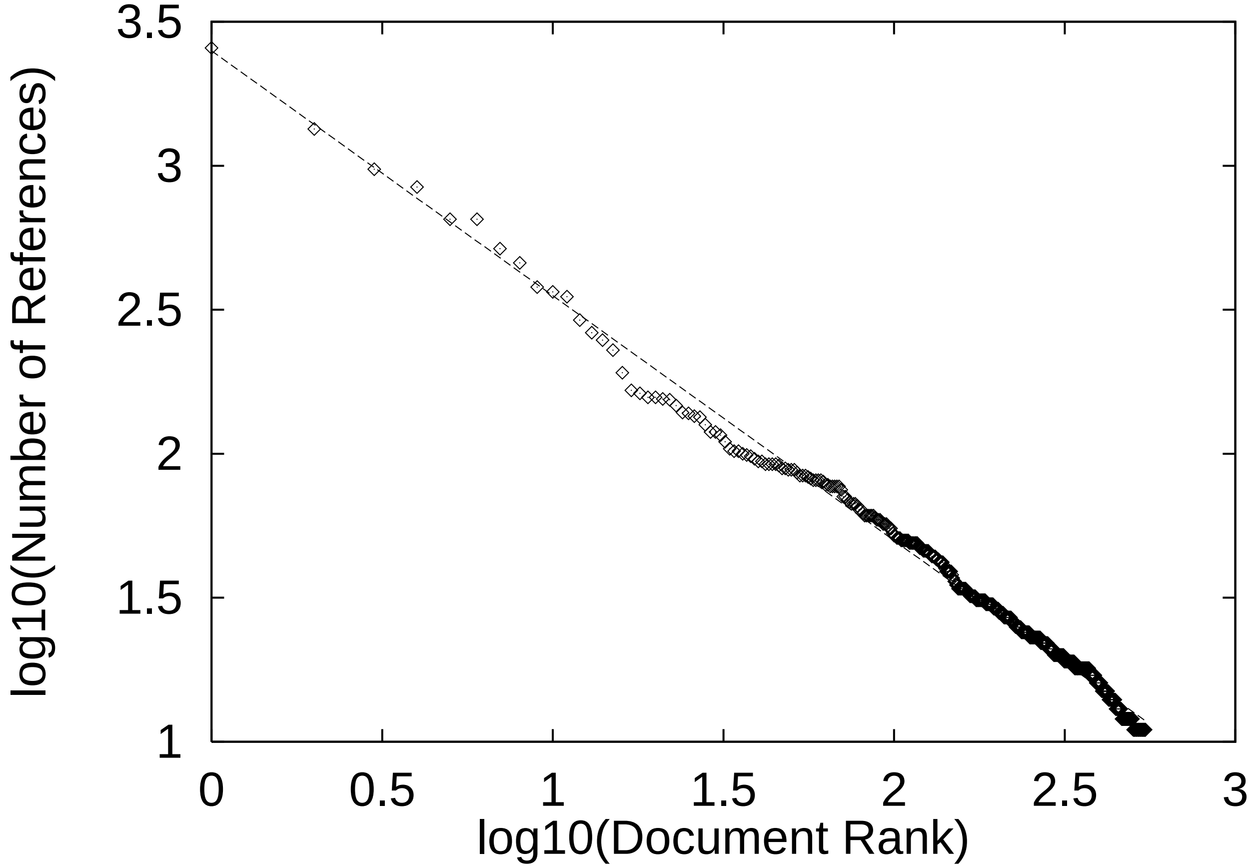
Log	NCSA	SDSC	EPA	BU
Duration	1 day	1 day	1 day	2 weeks
Start Date	Dec 19	Aug 22	Aug 29	Oct 08
Total requests	46,955	28,338	47,748	80,518
Unique requests	4,851	1,267	6,518	4,471

Summary of Access Log Data

```
ClientIP : TimeStamp :  RequestURL : Size
```

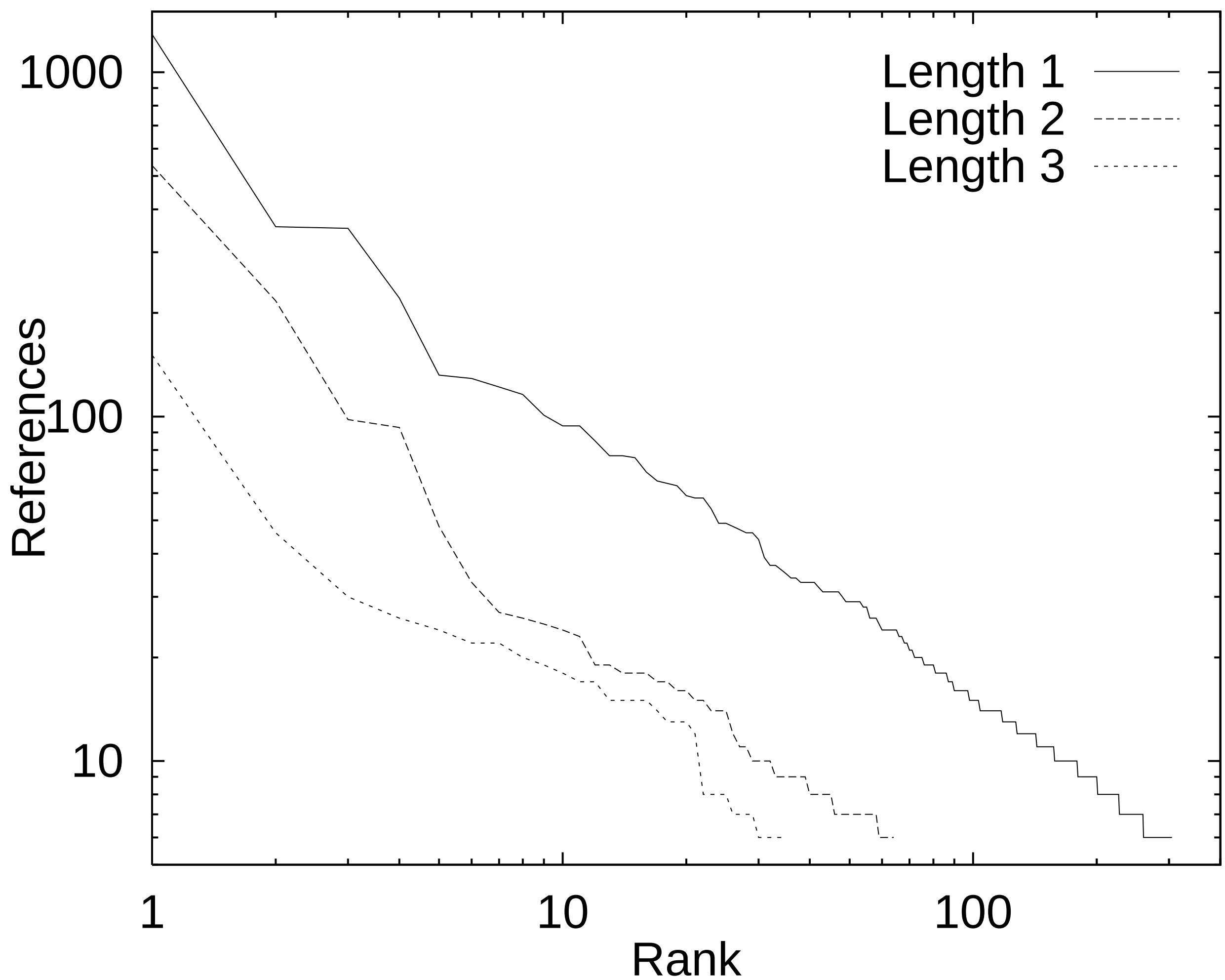
Data in a Typical Log Record

Characterizing Document Popularity



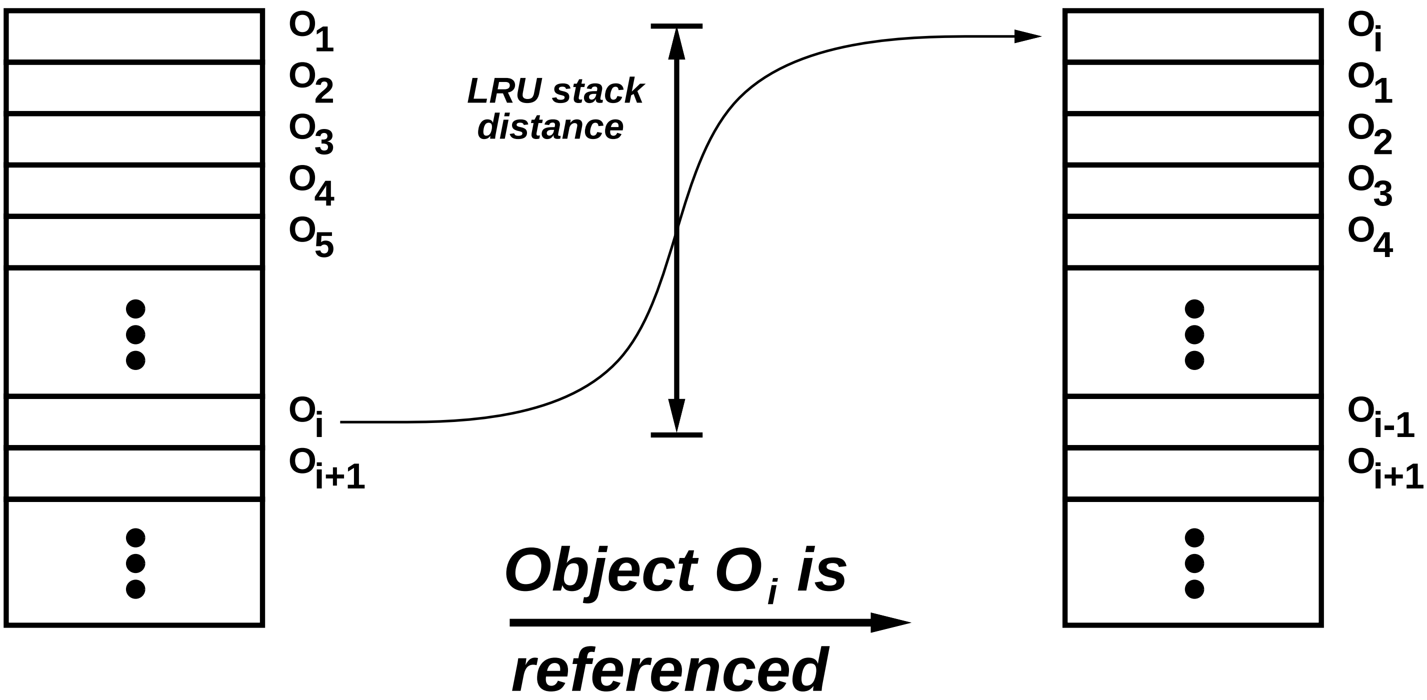
Zipf's Law Applied to Web Documents

Characterizing Document Popularity



Zipf's Law Applied to Sequences

Measuring Locality of Reference



LRU Stack Distance Model

Measuring Locality of Reference

- For any string of requests $R = r1.r2.r3\ldots$ we can compute a corresponding string of stack distances $D = d1.d2.d3\ldots$

- The request and distance strings are equivalent in terms of the locality of reference information they capture.

- The average stack distance of D is a measure of the number of intervening requests to unique objects between recurrring requests.

Evidence of Temporal Locality

- Consider a *scrambled* request string R' that corresponds to a random permutation of R.
- R and R' have the same Zipf popularity profile since they are permutations of each other.
- The difference in stack distance distribution for R and R' would be a measure of temporal locality.

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

Characterizing Temporal Locality

- If F_D is the distribution of the stack distance D, then the miss rate $M(C)$ of a cache that can hold C files is

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

  Knowledge of F_D provides enough information to predict the performance of a cache of any size for the given trace.

- Our analysis shows that F_D has a long tail, yet it does not seem to follow a power-law (*e.g.* Pareto).

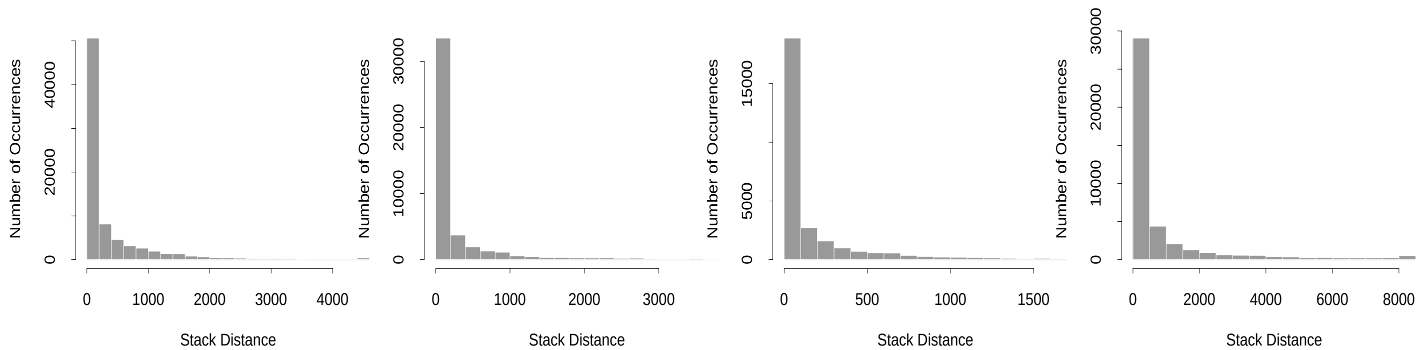
Characterizing Temporal Locality

- Lognormal distributions with parameters μ and σ seem to provide the best fit for the distributions of the stack distance in the traces we considered.

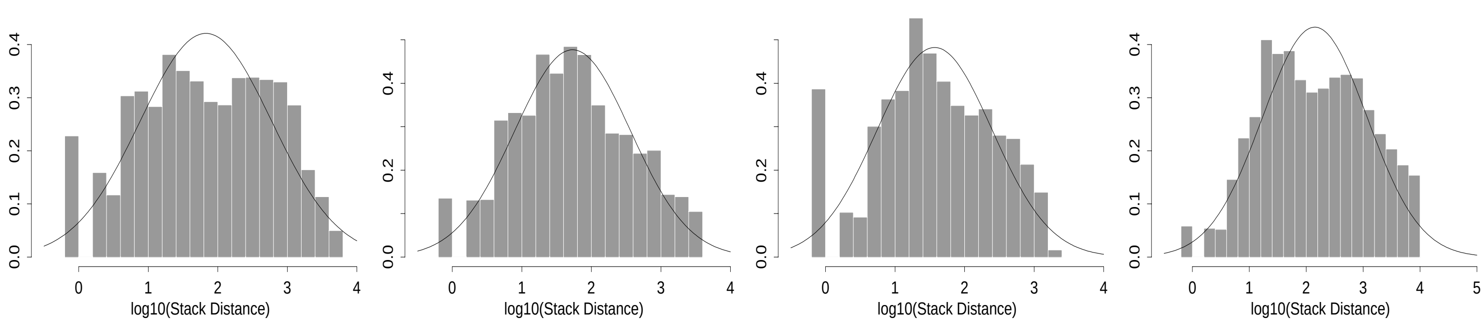
	BU	NCSA	SDSC	EPA
$\hat{\mu}$	1.829	1.730	1.568	2.150
$\hat{\sigma}$	0.947	0.836	0.827	0.921

Lognormal Distribution Parameters

Characterizing Temporal Locality

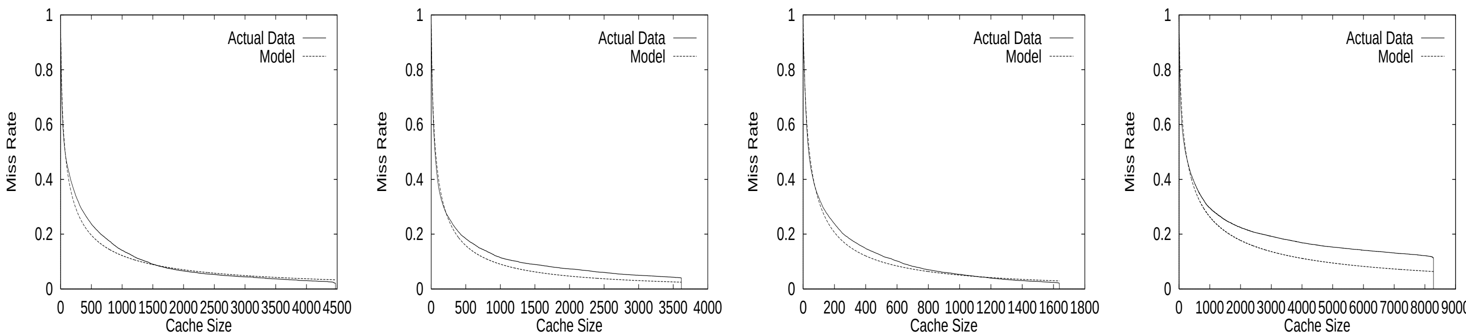


Stack Distance Distributions (BU, NCSA, SDSC, EPA)

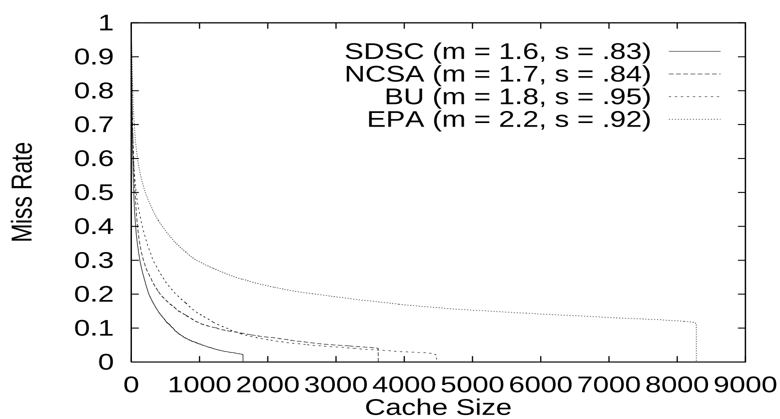


Lognormal Distributions and Fit (BU, NCSA, SDSC, EPA)

Temporal Locality Modeling Accuracy

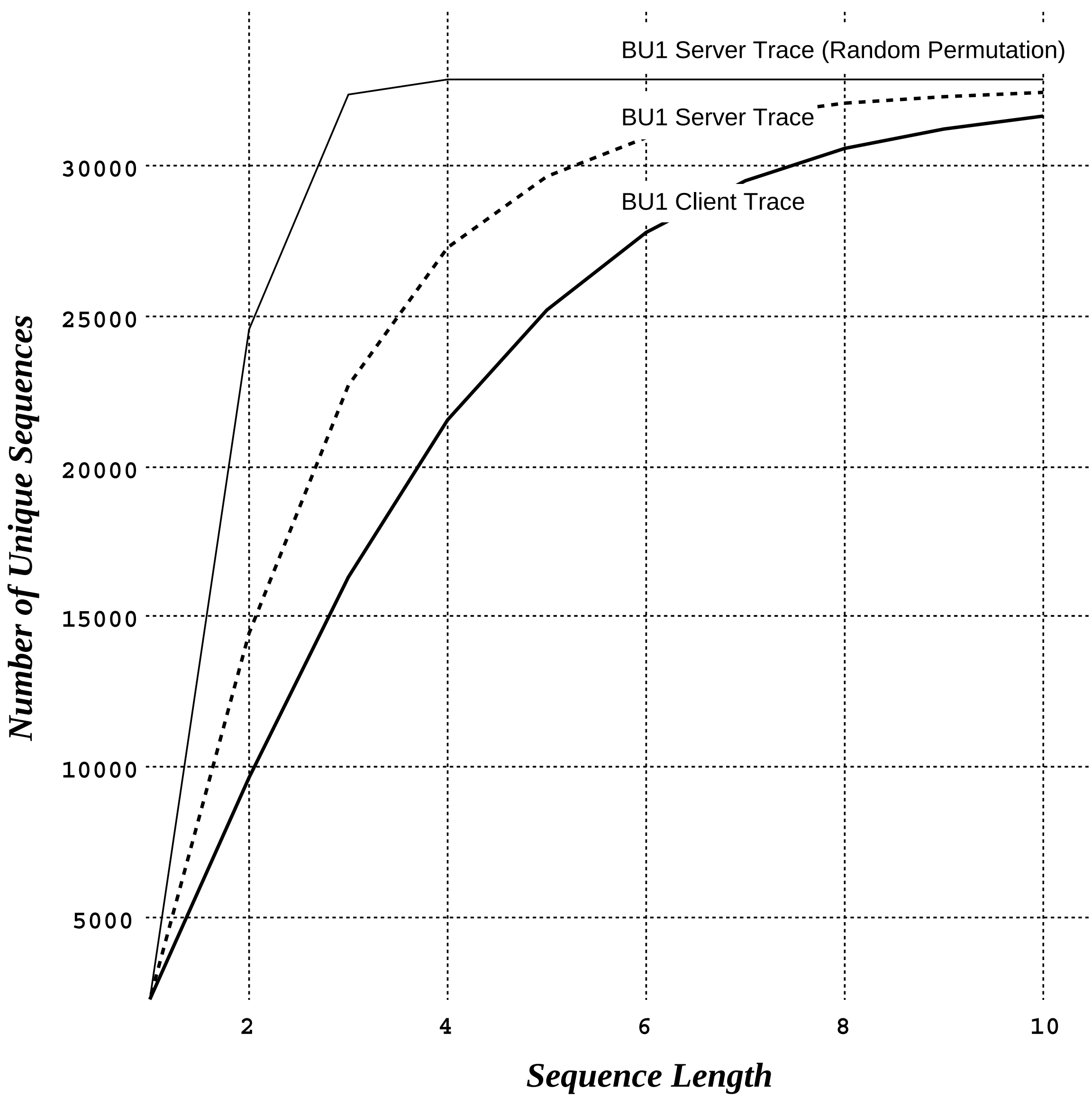


Actual and Predicted Miss Rates (BU, NCSA, SDSC, EPA)



Comparative Predicted Miss Rates (BU, NCSA, SDSC, EPA)

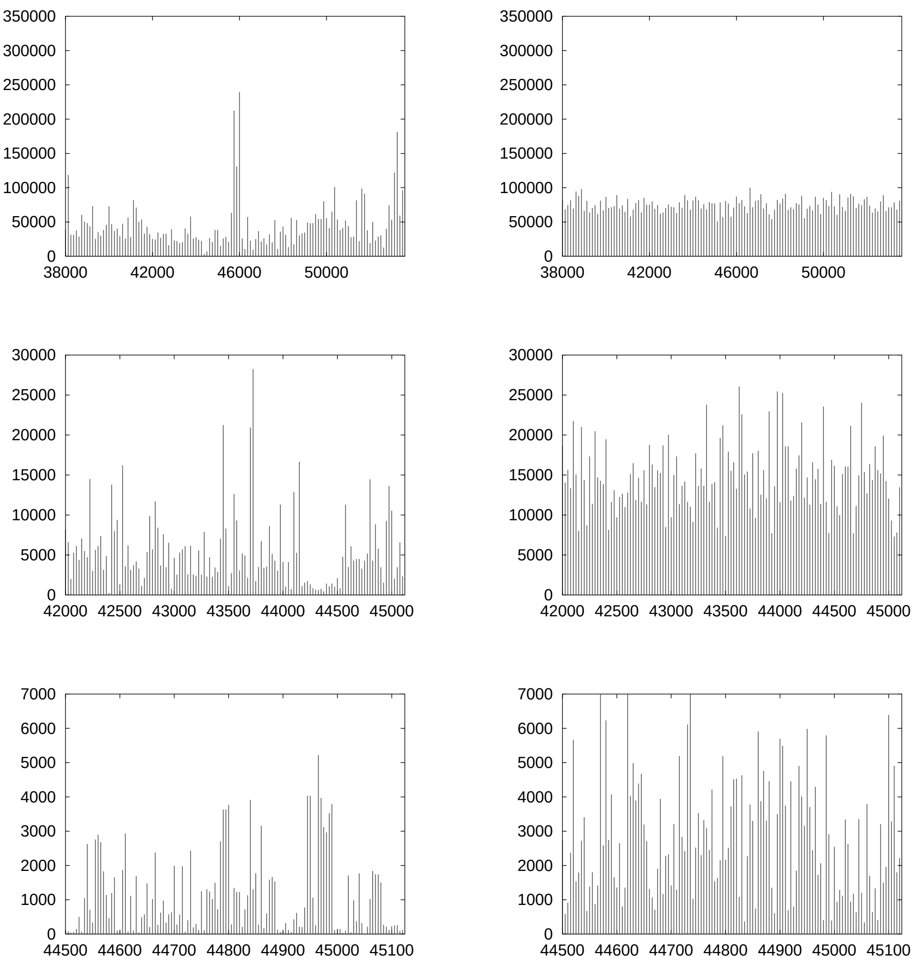
Evidence of Spatial Locality



Unique sequences observed in the BU trace

Characterizing Spatial Locality

- Stack distance series are bursty at all timescales
  $\rightsquigarrow$ They exhibit self-similar characteristics.



Stack Distance Scaling Behavior of Original (left) and Scrambled (right) BU Traces

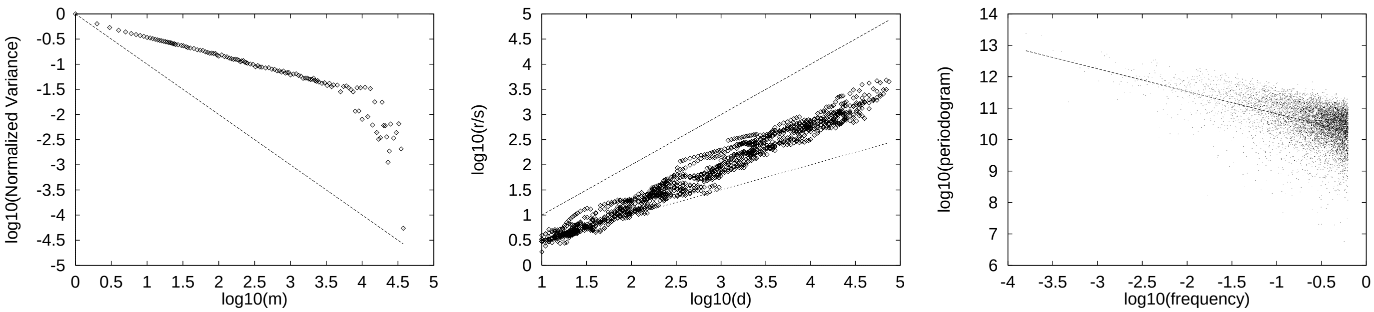
Characterizing Spatial Locality

- Stack distance self-similarity is evidence of very long-range correlations, which correspond to long periods of very large stack distances—caused by phase changes in referencing behavior.

- The degree of self-similarity is captured by the *Hurst* parameter H, which takes values between 0.5 and 1.0. As $H \rightarrow 1$, the burstiness becomes more pronounced at high levels of aggregation.

- We use four methods to estimate the H parameter for our datasets: the variance-time plot, the R/S plot, the periodogram, and the *Whittle* estimator, which provides confidence intervals as well.

Characterizing Spatial Locality



Graphical Estimators of H for BU Trace

	V-T	R/S	Per.	Wtl. (95% conf.)
BU	.82	.78	.87	.85 (0.84, 0.87)
NCSA	.71	.74	.74	.74 (0.73, 0.77)
SDSC	.71	.68	.69	.68 (0.66, 0.71)
EPA	.64	.66	.66	.65 (0.64, 0.67)

Estimates of H for Original Traces

Characterizing Spatial Locality

	Original Trace				Scrambled Trace			
	V-T	R/S	Per.	Wtl.	V-T	R/S	Per.	Wtl.
BU	.82	.78	.87	.85	.50	.55	.50	.50
NCSA	.71	.74	.74	.74	.50	.51	.51	.49
SDSC	.71	.68	.69	.68	.52	.54	.50	.50
EPA	.64	.66	.66	.65	.51	.55	.47	.50

H for Original *vs* Scrambled Traces

Synthetic Web Reference Trace Generation

Step 1 :

Select parameters μ and σ reflecting temporal locality, and H reflecting spatial locality—based on empirical measurement of traces to be imitated, or based on our results.

Step 2 :

Generate a stack distance trace with marginal distribution determined by μ and σ and long-range dependence determined by H using the two-phase approach described in [Huang *et al*: 1995].

Step 3 :

Invert the stack distance trace to form a sequence of file names.

Related Work

Traditional Memory Systems

- Fundamentals of reference locality in hierarchical memories [Denning and Schwartz: 1972].
- Stack distance analysis and algorithms [Mattson *et al*: 1970].
- Establish the existence of long-range dependence in reference strings [Spirn: 1976].
- Relate the fractal dimension of cache misses to software complexity [Voldman *et al*: 1983].
- Model memory access pattern as a random walk with fractal dimension [Thiebaut: 1989].

Related Work

Large-scale Information Systems

- Caching and replication for distributed file systems [Howard *et al*: 1988].
- Model Web access using Zipf-based popularity profiles [Glassman: 1994].
- Model Web access using frequency and recency rates of past accesses [Recker and Pitkow: 1994].
- Characteristics of client access patterns [Cunha, Bestavros, and Crovella: 1995].
- Used Markov processes to model access inter-dependencies [Cunha and Bestavros: 1995, 1996].

Current and Future Work

- Incorporate file size information in the access pattern characterization.
- Study the effect of increased multiprogramming levels on access pattern characteristics.
- Study the implication on caching and prefetching algorithms at clients and servers.
- Use measured characteristics to design benchmarks for evaluating client and server software.