Locality Approximation Using Time

Xipeng Shen

College of William and Mary

Williamsburg, VA, USA

Jonathan Shaw

Shaw Technologies

Portland, OR, USA

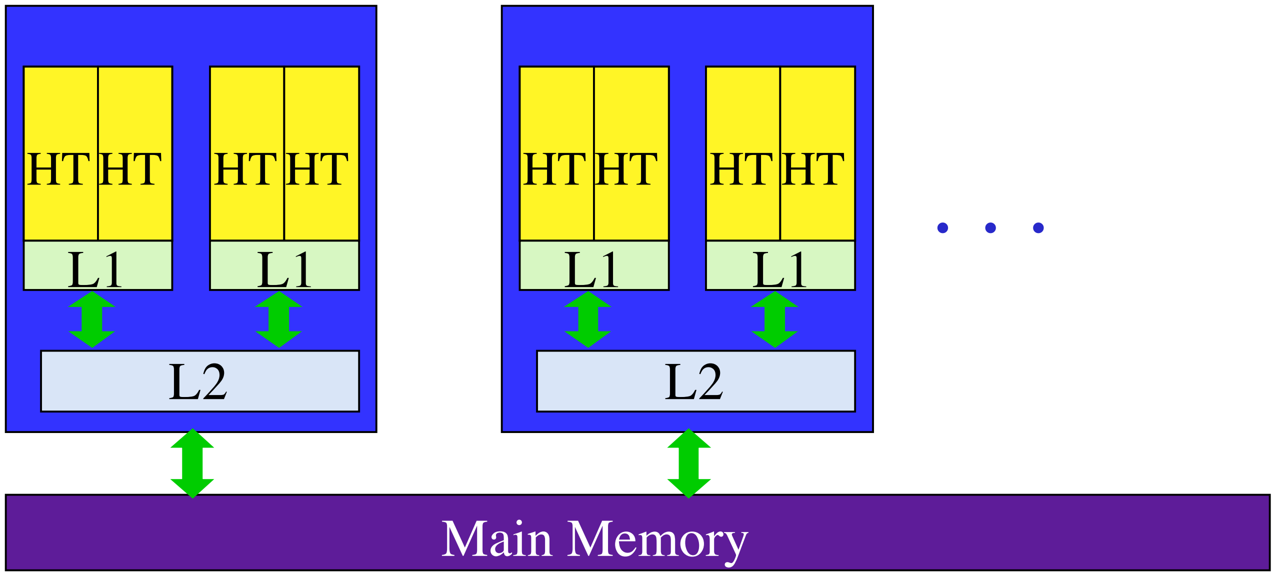
Brian Meeker and Chen Ding

University of Rochester

Rochester, NY, USA

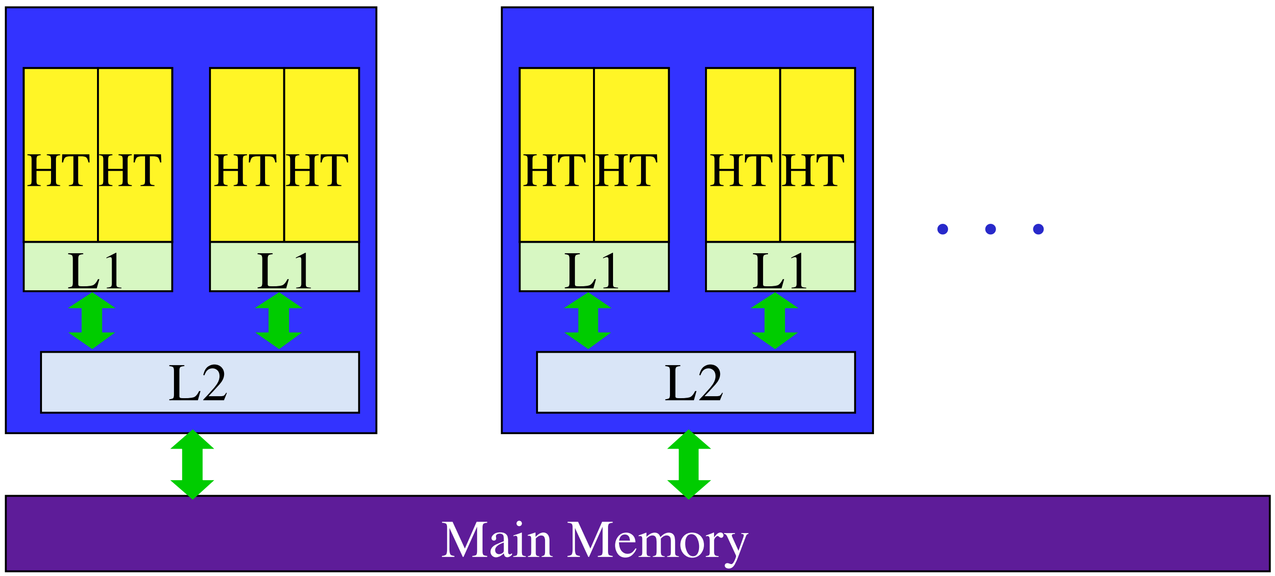
Introduction

- Heterogeneous cache sharing



Introduction

- Heterogeneous cache sharing



Locality analysis should be flexible to handle different caches and contentions

Introduction

- Contentions cause dynamic changes of available cache resources.
- Applications' needs also change dynamically.

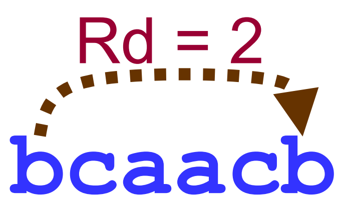
Introduction

- Contentions cause dynamic changes of available cache resources.
- Applications' needs also change dynamically.

Locality analysis should be efficient to measure, ideally being applicable during runtime.

Reuse Distance: A Flexible Locality Model

- Definition [Mattson et. al. 1970, Ding+ 2003]
  - the number of distinct elements between this and the previous access to the same data



- Comparison

Reuse Distance: A Flexible Locality Model

- Definition [Mattson et. al. 1970, Ding+ 2003]
  - the number of distinct elements between this and the previous access to the same data

- Comparison

Reuse Distance: A Flexible Locality Model

- Definition [Mattson et. al. 1970, Ding+ 2003]
  - the number of distinct elements between this and the previous access to the same data

- Comparison

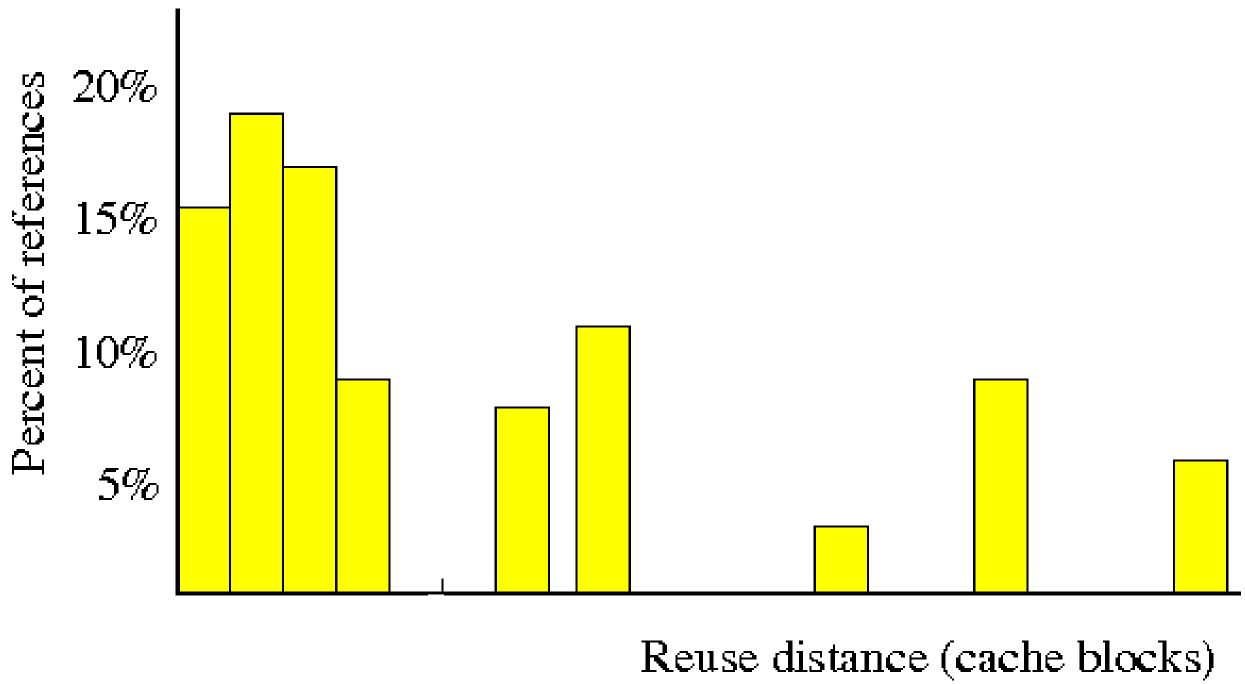
Reuse Distance

- hardware independent
- defined on each point, no need for windows
- a distance at each access

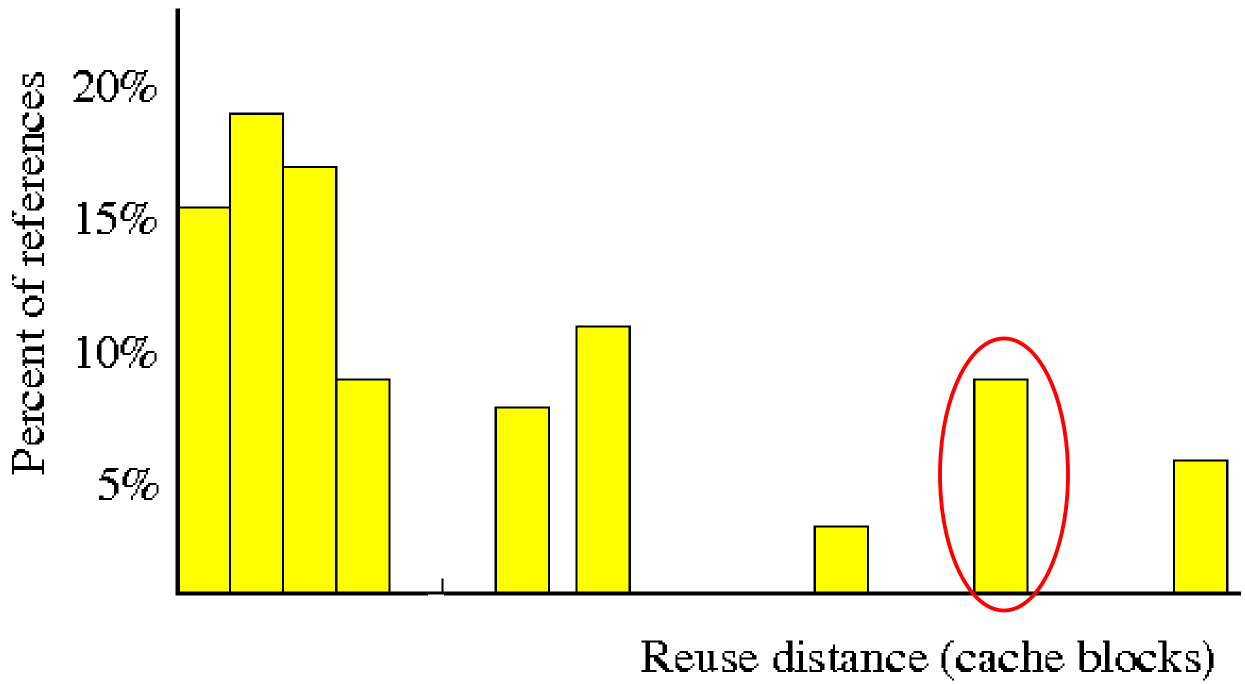
Cache Miss Rate

- hardware dependent
- interval-based, defined on windows
- hit or miss at each access

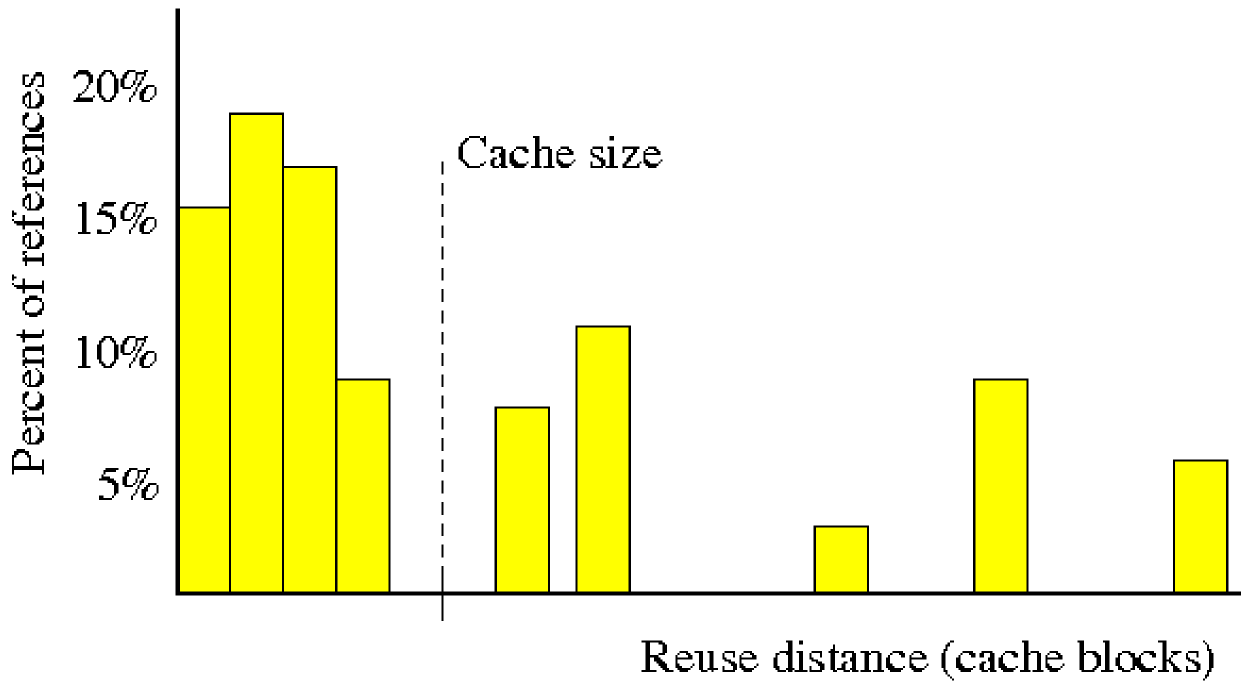
Reuse Distance Histogram



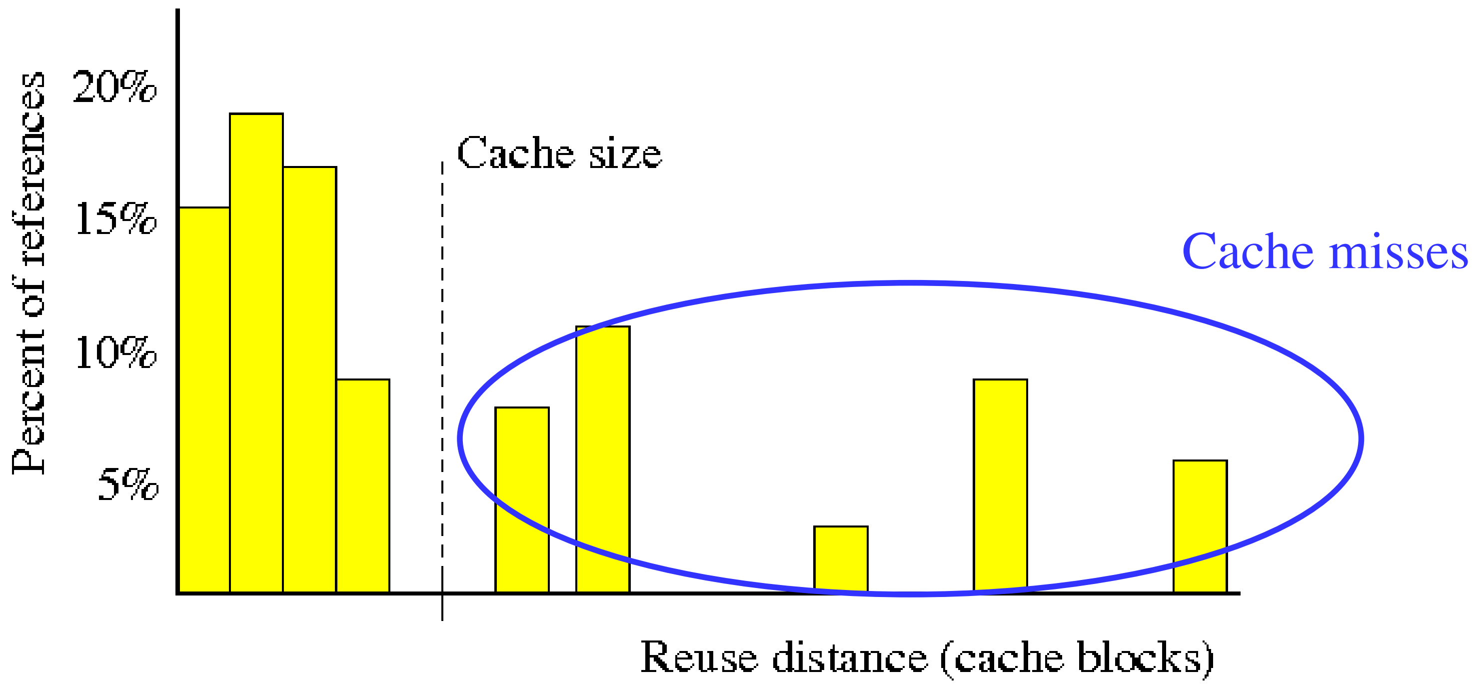
Reuse Distance Histogram



Reuse Distance Histogram



Reuse Distance Histogram



Many Uses of Reuse Distance in Research

- Study cache reuses *[Ding+:SC04,Huang+:ASPLOS05]*
- Guide and evaluate program transformation *[Almasi+:MSP02,Zhong+:TOC07]*
- Discover locality-improving refactoring *[Beyls+:HPCC06]*
- Model cache sharing *[Chandra+:HPCA05]*
- Insert cache hints *[Beyls+:JSA05]*
- Manage superpages *[Cascaval+:PACT05]*
- Guide memory disambiguation *[Fang+:PACT05]*
- Predict program performance *[Marin+:SIGMETRICS04]*
- Model reference affinity *[Zhong+:PLDI04]*
- Detect locality phases *[Shen+:ASPLOS04]*

Few if Any Uses in Practice

The Big Hurdle

- High cost of measurement
  - T: execution length;   N: data size

1970 (Mattson+)
1975 (Bennett+)
1981 (Olken)
1991 (Kim+)
1993 (Sugumar+)
2002 (Almasi+)
2003 (Ding+)

O (T*N) → O (T**loglog*N)

The Big Hurdle

- High cost of measurement
  - T: execution length;   N: data size

1970 (Mattson+)
1975 (Bennett+)
1981 (Olken)
1991 (Kim+)
1993 (Sugumar+)
2002 (Almasi+)
2003 (Ding+)

O (T*N)

O (T**loglog*N)

But still, measuring a **1-minute** run takes several **hours!**

Our Solution

- Previous methods
  - Essentially implement the definition of reuse distance:

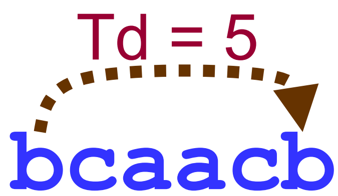
    "Counting" the # of distinct data
- Rationale of our method:
  - Use some "cheap" program behavior to statistically approximate reuse distance

The "Cheap" Behavior: Time Distance

- Time distance definition: the number of elements between this and the previous access to the same data.

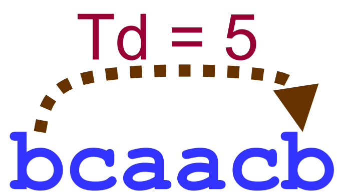
The "Cheap" Behavior: Time Distance

- Time distance definition: the number of elements between this and the previous access to the same data.

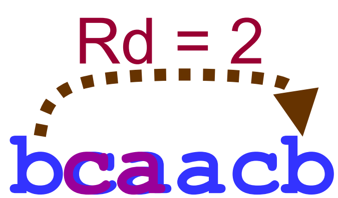


The "Cheap" Behavior: Time Distance

- Time distance definition: the number of elements between this and the previous access to the same data.



- Reuse distance definition: the number of distinct elements between this and the previous access to the same data.



From Time Distance to Reuse Distance

- Is it possible?

From Time Distance to Reuse Distance

- Is it possible?

b....b

Td=5 ⟹ Rd= 0, 1, 2, 3, or 4?

From Time Distance to Reuse Distance

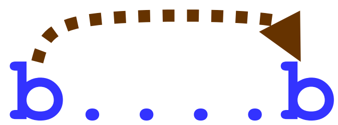
- Is it possible?

b....b

Td=5 ⟹ Rd= 0, 1, 2, 3, or 4? *No idea.*

From Time Distance to Reuse Distance

- Is it possible?



Td=5 ⟹ Rd= 0, 1, 2, 3, or 4? *No idea.*

- What if given the time distance histogram?
  - E.g. totally 4 reuses; one time distance is 5, three are all 1.

From Time Distance to Reuse Distance

- Is it possible?

b....b

Td=5 ⟹ Rd= 0, 1, 2, 3, or 4? *No idea.*

- What if given the time distance histogram?
  - E.g. totally 4 reuses; one time distance is 5, three are all 1.

bxxxxb

From Time Distance to Reuse Distance

- Is it possible?

  b....b

  Td=5 ⟹ Rd= 0, 1, 2, 3, or 4? *No idea.*

- What if given the time distance histogram?
  - E.g. totally 4 reuses; one time distance is 5, three are all 1.

  bxxxxb

  Td=5 ⟹ Rd= 1.

From Time Distance to Reuse Distance

- Is it possible?

b....b

Td=5 ⟹ Rd= 0, 1, 2, 3, or 4?

No idea.

- What if given the time distance histogram?
  - E.g. totally 4 reuses; one time distance is 5, three are all 1.

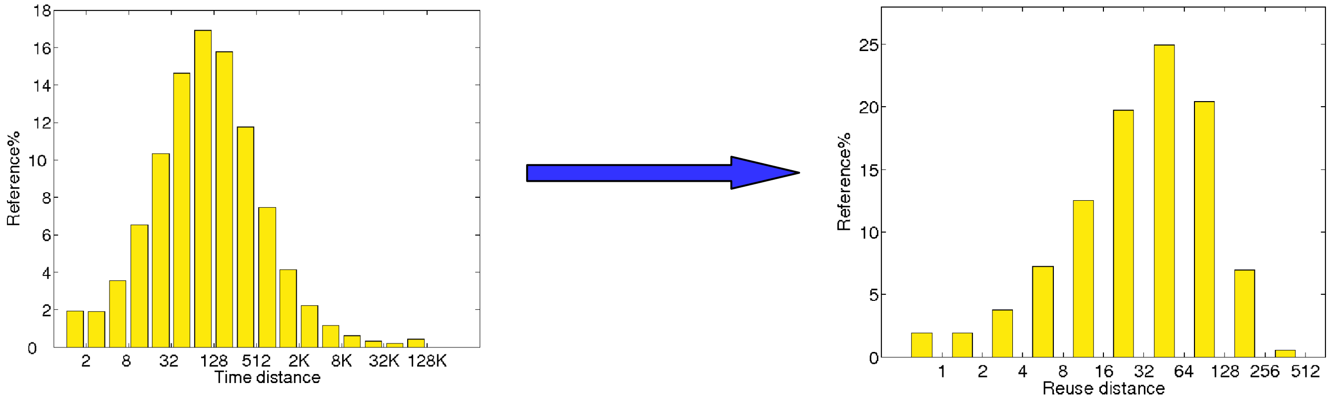
bxxxxb

Td=5 ⟹ Rd= 1.

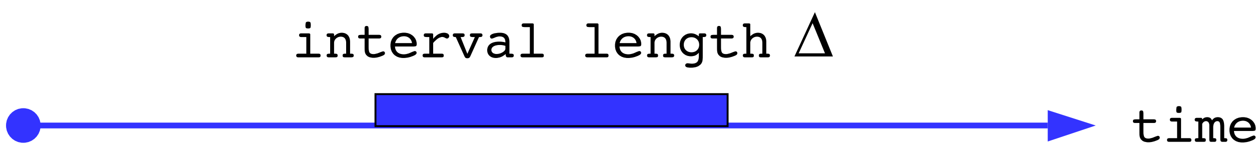
Okey. Generalize?

General Form of the Problem

Time distance histogram　　　Reuse distance histogram



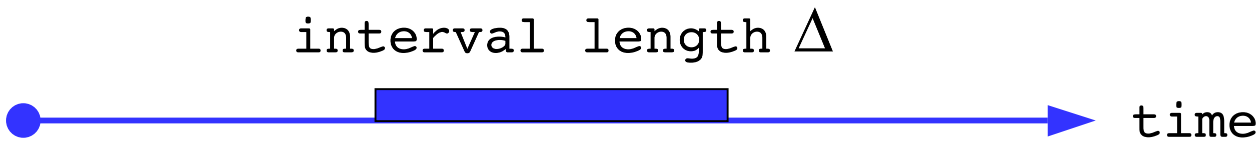
Outline of the Statistical Model



$P_3(\Delta)$:the expected probability for a variable to appear in the interval.

- Given $P_3(\Delta)$, the problem becomes a *Bernoulli* process (like tossing *N* coins.)
  -

Outline of the Statistical Model

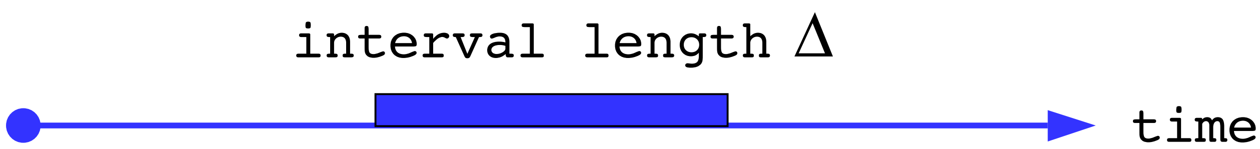


$P_3(\Delta)$: the expected probability for a variable to appear in the interval.

- Given $P_3(\Delta)$, the problem becomes a *Bernoulli* process (like tossing *N* coins.)
  - The probability of having *k* distinct variables in a Δ-long interval:

$$P(k,\Delta) = \binom{N}{k} (P_3(\Delta))^k (1 - P_3(\Delta))^{N-k}$$

Outline of the Statistical Model



$P_3(\Delta)$: the expected probability for a variable to appear in the interval.

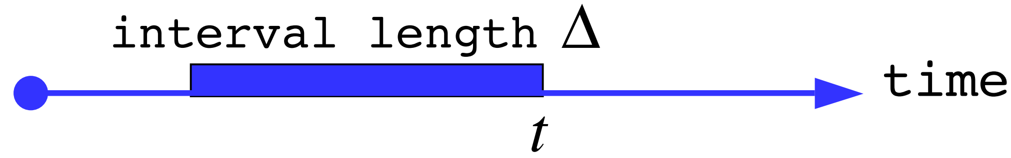
- Given $P_3(\Delta)$, the problem becomes a *Bernoulli* process (like tossing *N* coins.)
  - The probability of having *k* distinct variables in a Δ-long interval:

$$P(k,\Delta) = \binom{N}{k} (P_3(\Delta))^k (1 - P_3(\Delta))^{N-k}$$

Enough to calculate the reuse distance histogram.

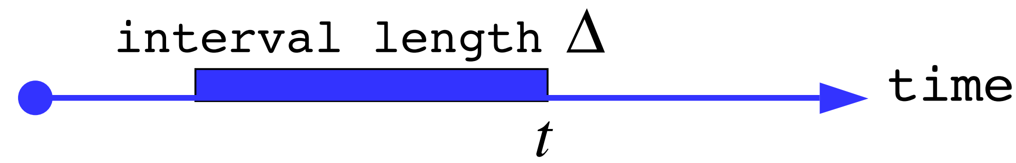
Outline of the Statistical Model

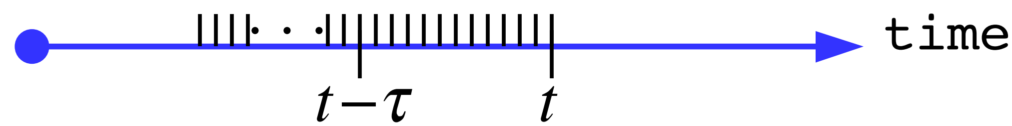
- $P_3(\Delta)$: the expected probability for a variable to appear in Δ.



Outline of the Statistical Model

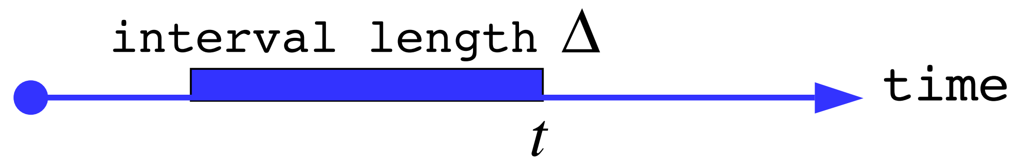
- $P_3(\Delta)$: the expected probability for a variable to appear in Δ.





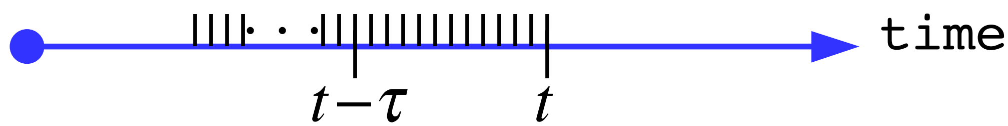
Outline of the Statistical Model

- $P_3(\Delta)$: the expected probability for a variable to appear in Δ.



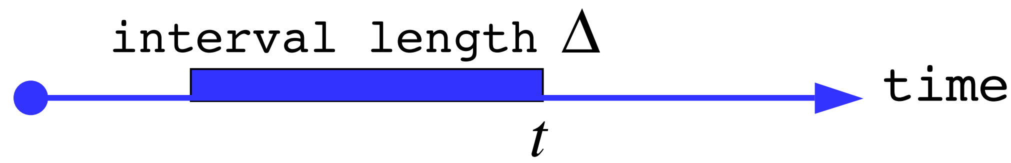
- $P_2(\tau)$: the probability for a variable's last access before t to be at time $(t-\tau)$.

$$\therefore P_3(\Delta) = \sum_{\tau=1}^{\Delta} P_2(\tau)$$



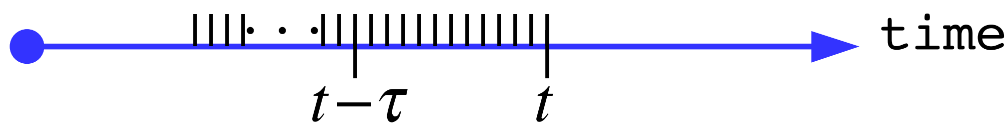
Outline of the Statistical Model

- $P_3(\Delta)$: the expected probability for a variable to appear in Δ.



- $P_2(\tau)$: the probability for a variable's last access before t to be at time $(t-\tau)$.

$$\therefore P_3(\Delta) = \sum_{\tau=1}^{\Delta} P_2(\tau)$$



- The following is proved:

$$P_2(\tau) = \sum_{\varepsilon=\tau+1}^{T} \frac{1}{N} P_T(\varepsilon)$$

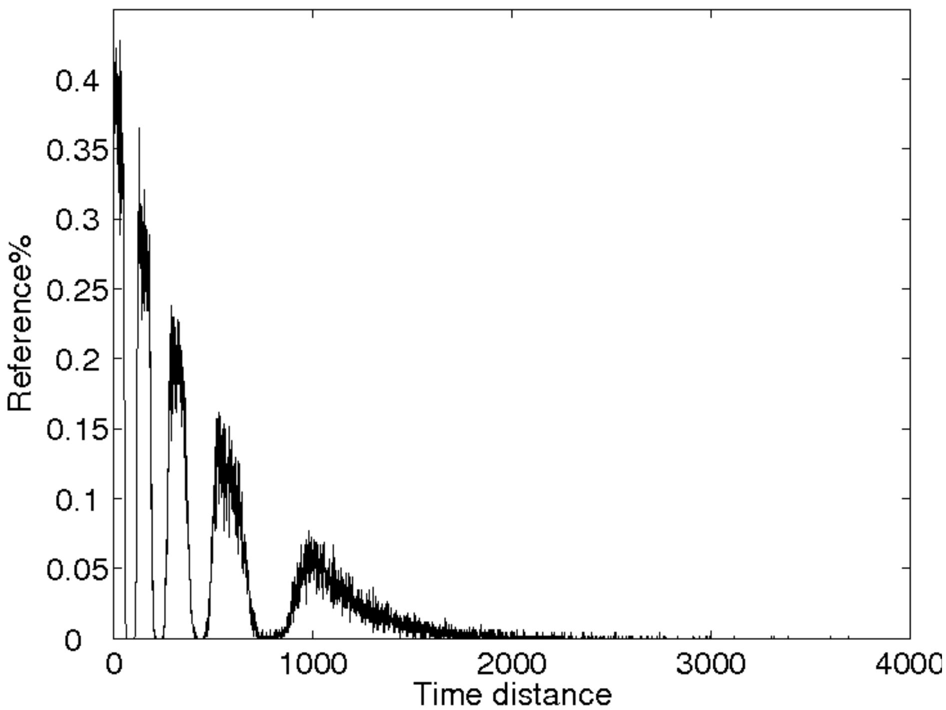
where $P_T(\varepsilon)$ is the value of the time distance histogram when $Td = \varepsilon$.

Methodology

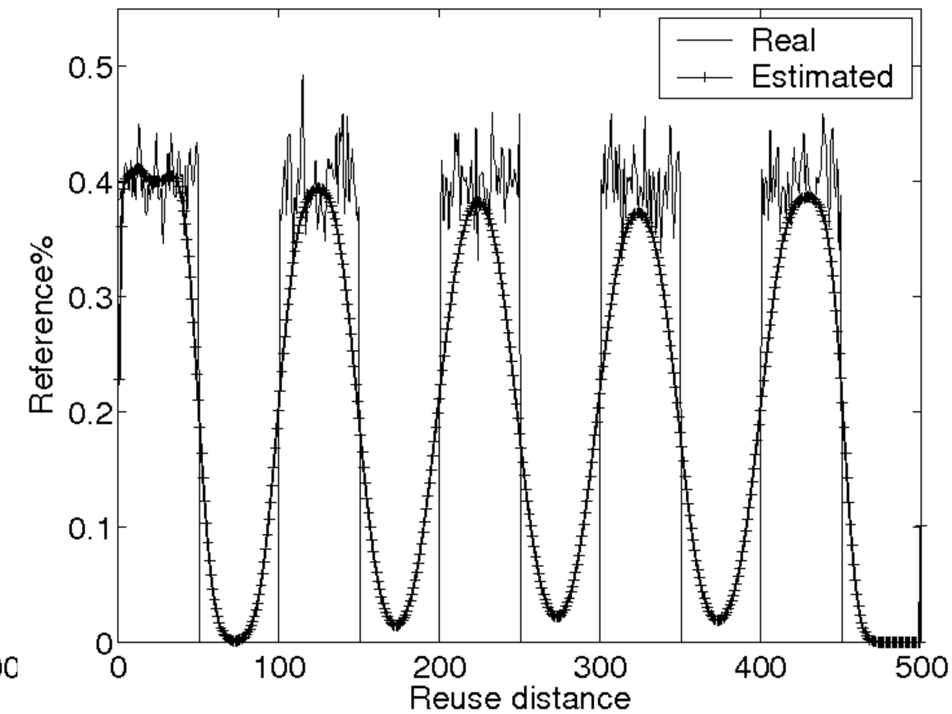
- Implementation
  - PIN 3.4 for instrumentation
  - GCC -O3
  - Intel Xeon 2GHz, Fedora Core 3 Linux
- Comparison
  - Ding and Zhong's technique [PLDI'03]
    - Asymptotically the fastest tool for measuring reuse distance

Evaluation (Pulse-like reuse distributions)

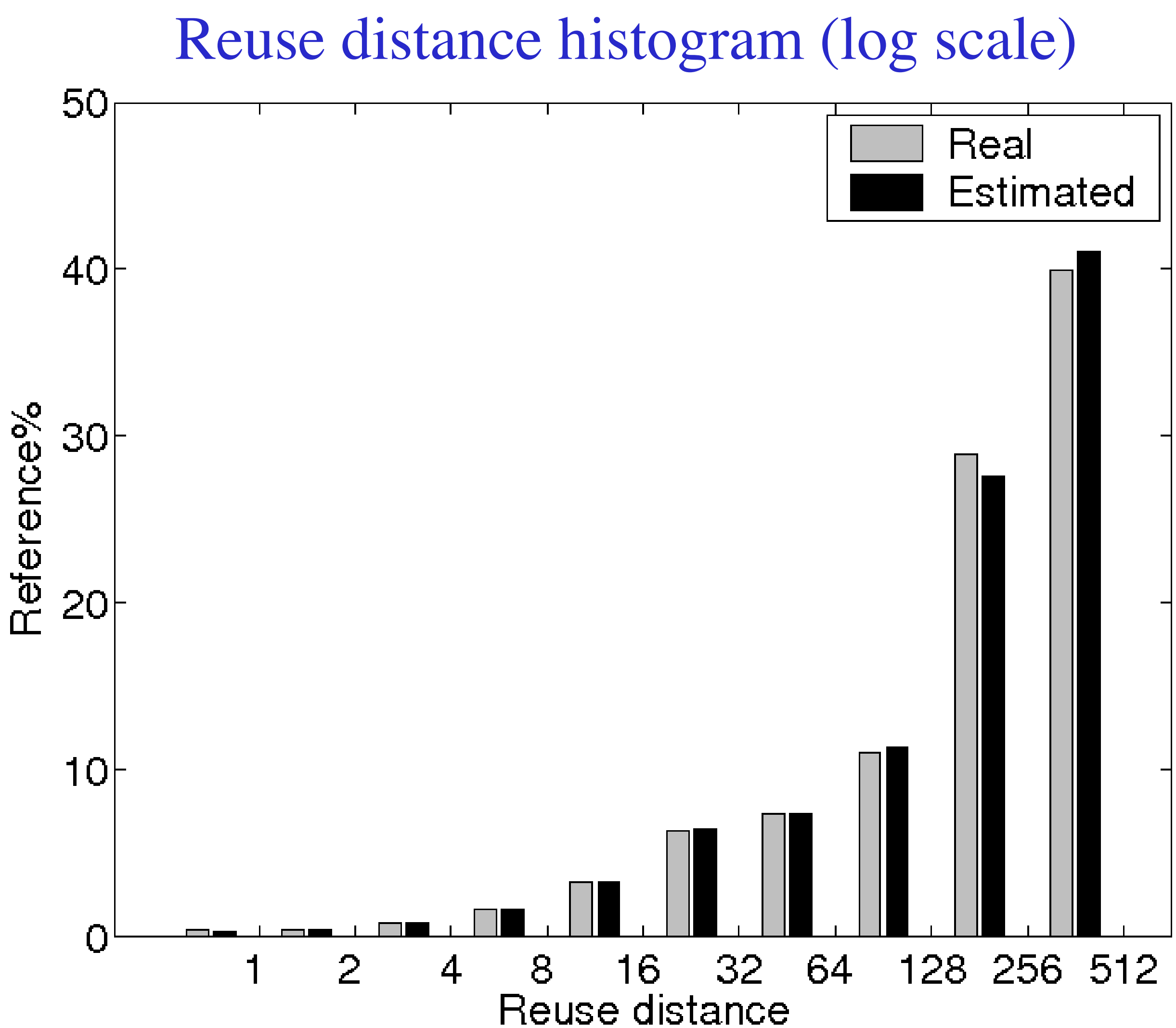
Time distance histogram



Reuse distance histogram (linear scale)



Evaluation (Pulse-like reuse distributions)



Evaluation (SPEC CPU2000)

Benchmarks		Speedup	Accuracy
CINT	crafty	15.6X	100
	gcc	10.1X	99.9
	gzip	12.3X	99.6
	mcf	19.2X	98.4
	twolf	17.4X	97.5
	vortex	15.6X	100
CFP	ammp	19.5X	99.5
	applu	19.9X	99.4
	equake	21.1X	98.5
	mesa	18.7X	100
	mgrid	21.5X	99.8
	wupwise	15.6X	99.6
Average		**17.2X**	**99.4**

Summary

- Demonstrate the connection between time and locality
- Propose a probabilistic model for fast accurate reuse distance approximation
- A step toward practical uses with potential for runtime uses

Thanks!