

HOTL: A Higher Order Theory of Locality

Chen Ding

University of Rochester

with Xiaoya Xiang, Bin Bao and Hao Luo

The Story of a Theory

- An empirical observation
 - “During any interval of execution, a program favors a subset of its pages, and this set of favored pages changes slowly”
-- Peter Denning
 - The 80-20 rule
 - The law of diminishing returns
- How to quantify?
 - the locality of a program or an operation?
 - what are “primary” metrics?
- Two example quantities
 - data volume and reuse time
 - the connection is the key

Locality Statistics I

Miss Ratio



WIKIPEDIA
The Free Encyclopedia

Article

Discussion

Read

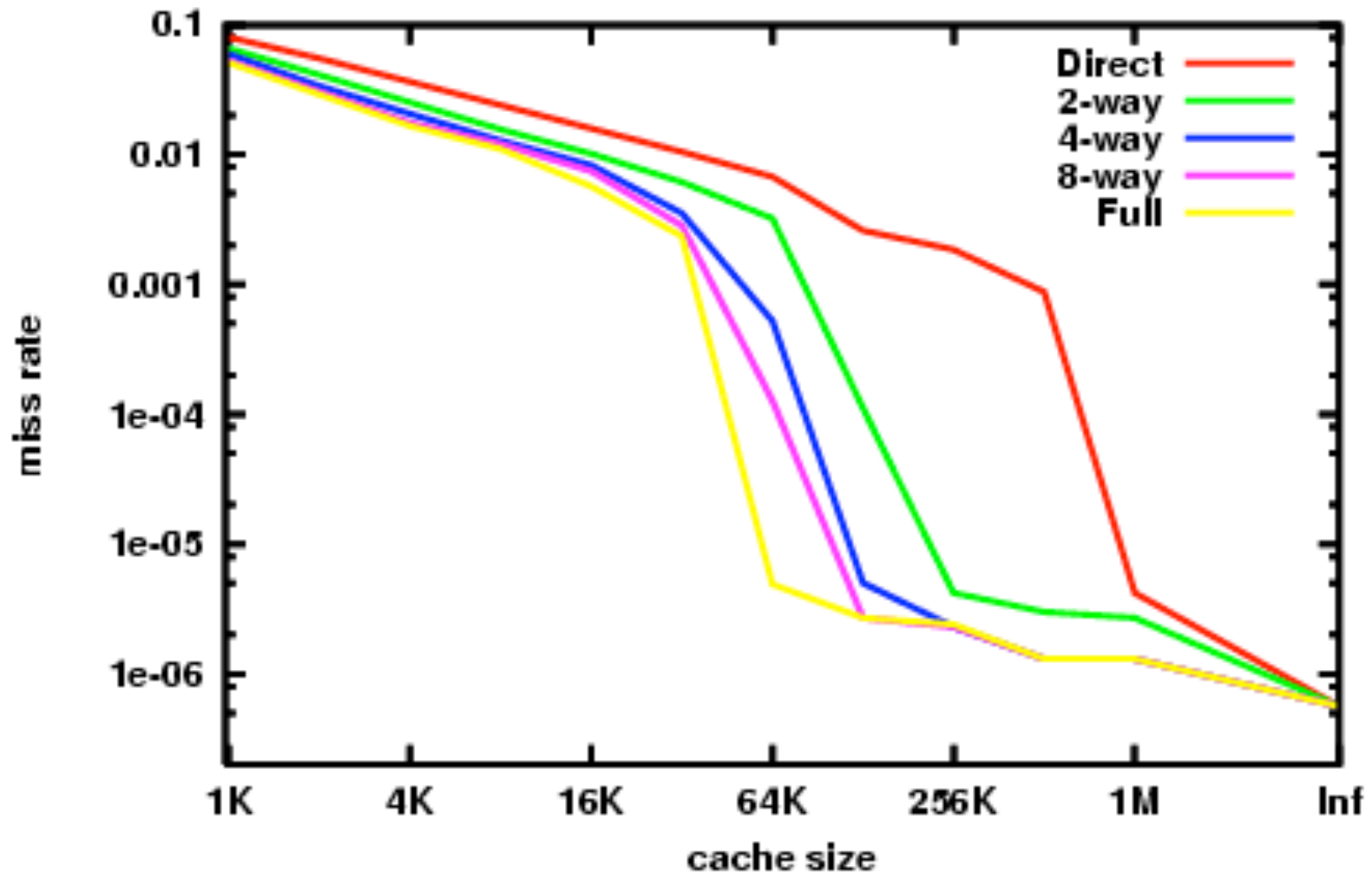
Edit

View history

Search

CPU cache

From Wikipedia, the free encyclopedia



Cache Performance for SPEC CPU2000 Benchmarks

Version 3.0

May 2003

Jason F. Cantin

Department of Electrical and Computer Engineering

1415 Engineering Drive

University of Wisconsin-Madison

Madison, WI 53706-1691

jcantin@ece.wisc.edu

<http://www.jfred.org>

Mark D. Hill

Department of Computer Science

1210 West Dayton Street

University of Wisconsin-Madison

Madison, WI 53706-1685

markhill@cs.wisc.edu

<http://www.cs.wisc.edu/~markhill>

<http://www.cs.wisc.edu/multifacet/misc/spec2000cache-data>



WIKIPEDIA
The Free Encyclopedia

Article

Discussion

Read

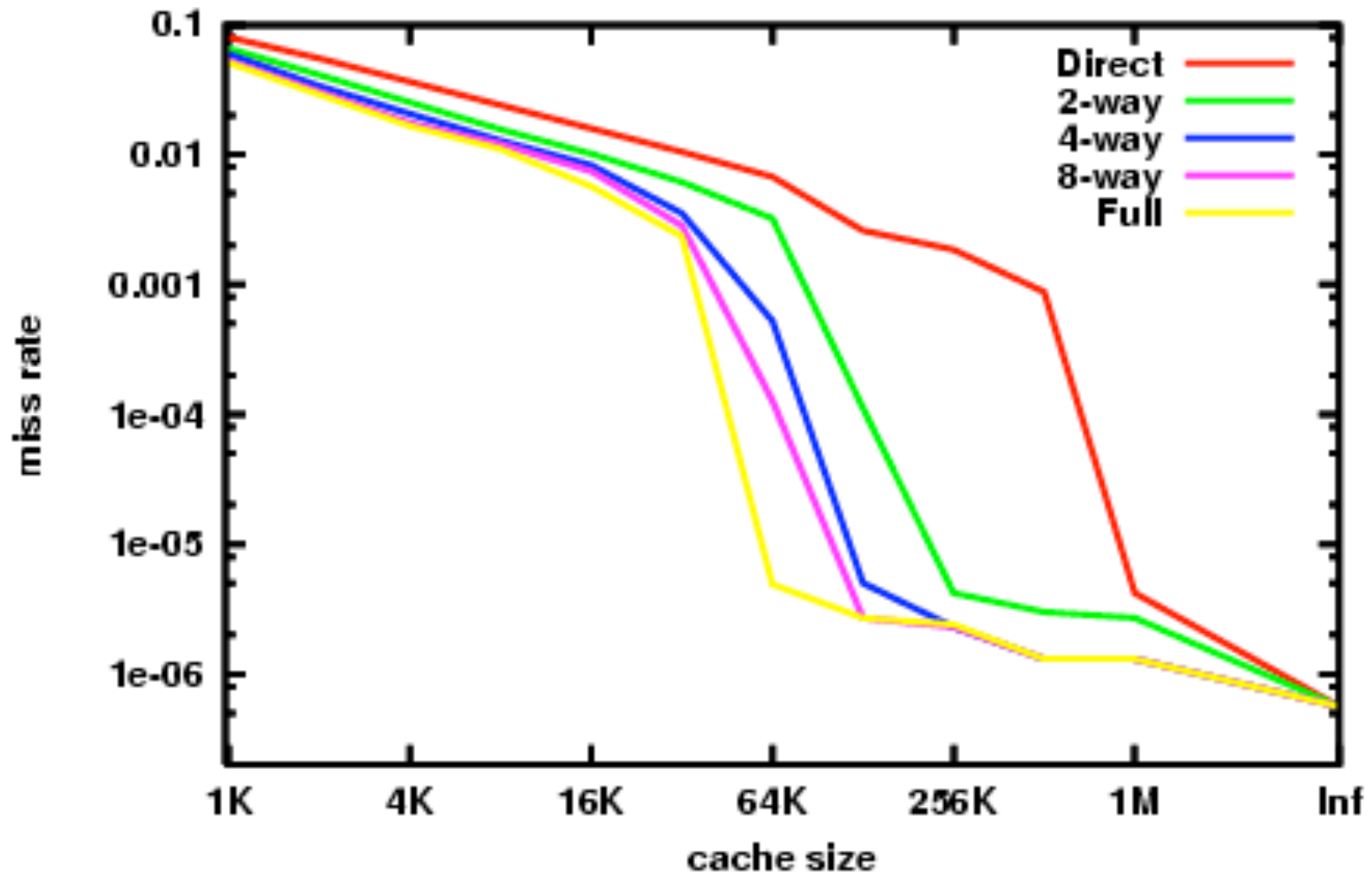
Edit

View history

Search

CPU cache

From Wikipedia, the free encyclopedia



Locality Statistics II

Reuse Distance

(i.e. LRU stack distance [Mattson et al. IBM 1970])

Reuse Distance

- Reuse distance of an access to x
 - the number of distinct data accessed after the previous access to x
- Reuse distance (rd) vs reuse time (rt)
- Distribution function $P(rd)$ and $P(rt)$
 - the distribution of all rd and rt
 - $P(rd=3) = 2/7$
 - $P(rt=3) = 0/7$
- $P(rd)$ determines the LRU miss ratio
 - $mr(3) = P(rd>3) = 3/7$
 - Smith formula to account for associativity
- Not enough to model cache sharing

rt :	∞	∞	∞	4	2	4	6
rd :	∞	∞	∞	3	1	2	3
	a	b	c	a	a	c	b

Locality Statistics III

Footprint

[Ding, Xiang et al. PPOPP 2008/11, PACT 11]

Footprint

- Amount of data access in an execution period
 - $fp(w)$: average footprint of ALL windows of length w
 - length- n trace, $O(n^2)$ windows
 - 1 billion accesses, half quintillion windows
- Example: "abbb"
 - 3 length-2 windows: "ab", "bb", "bb"
 - footprints 2, 1, 1
 - the average $fp(2) = (2 + 1 + 1)/3 = 4/3$
- Example "xyz xyz ..."
 - $fp(i) = i$ for $0 \leq i \leq 3$
 - $fp(i) = 3$ for $i > 3$

Footprint Measurement

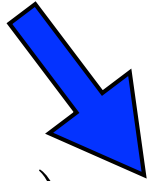
- Working set
 - limit value in an infinitely long trace [Denning & Schwartz 1972]
- Direct counting
 - single window size [Thiebaut & Stone TOCS'87]
 - seminal paper on footprints in shared cache
 - same starting point [Agarwal & Hennessy TOCS'88]
- Statistical approximation
 - [Denning & Schwartz 1972; Suh et al. ICS'01; Berg & Hagersten PASS'04; Chandra et al. HPCA'05; Shen et al. POPL'07]
 - level of precision couldn't be quantified
- Recent precise definition/solutions [Xiang+ PPOPP'11, PACT'11]
 - footprint distribution, $O(n \log m)$ [Xiang et al. PPOPP'11]
 - footprint function, $O(n)$ [Xiang et al. PACT'11]

Higher-order Locality Theory

HOTL Conversion

The Xiang formula for fp [PACT'11]

- **rt**: reuse time
- **m**: data size
- **n**: trace length

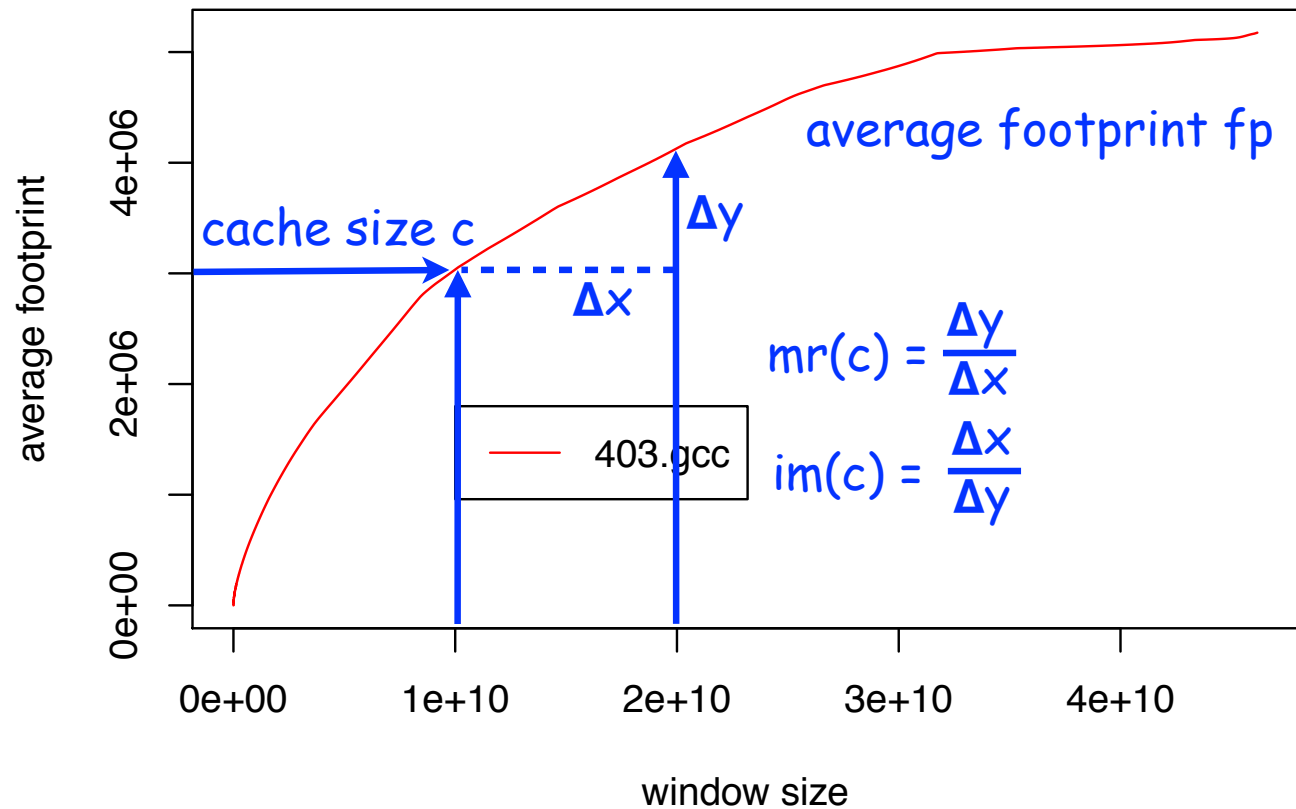

$$fp(x) \approx m - \sum_{k=x+1}^{n-1} (k - x) P(rt = k)$$

$$mr(c) = mr(fp(x)) = \frac{fp(x + \Delta x) - fp(x)}{\Delta x}$$

$$P(rd = c) = mr(c - 1) - mr(c)$$

fp to mr Conversion

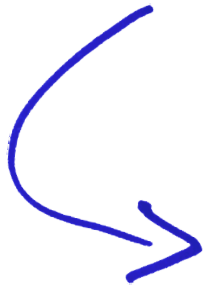
- Δx
 - average time for Δy misses
- $\frac{\Delta y}{\Delta x}$
 - miss rate at size c



Two HOTL Theories

HOTL hierarchy	working set locality theory (WSLT)	cache locality theory (CLT)
data volume (3rd order)	mean WS size $s(T)$	mean footprint $fp(T)$, mean fill time $vt(c)$
miss rate (2nd order)	time-window miss rate $m(T)$, lifetime $L(T)=1/m(T)$	LRU miss rate $mr(c)$, inter-miss time $im(c)=1/mr(c)$
reference behavior (1st order)	inter-reference interval (reuse time) distribution $P(iri=x)$	reuse distance distribution $P(rd=x)$

Denning & Schwartz, 1972



Denning's Law of Locality *The working set is the second-order sum of the reuse frequency, and conversely, the reuse frequency is the second-order difference of the working set.*

HOTL conversion formulas

$$fp(x) \approx m - \sum_{k=x+1}^{n-1} (k - x)P(rt = k)$$

$$mr(c) = mr(fp(x)) = \frac{fp(x+\Delta x) - fp(x)}{\Delta x}$$

$$P(rd = c) = mr(c - 1) - mr(c)$$

trace “xyzxyz...”

t	$fp(t)$	c	$mr(c)$	$P(rd=c)$
1	1	1	1	0
2	2	2	1	0
3	3	3	0	1
4	3	4	0	0

But is it correct in general?

HOTL for Cache

Xiang formula (any trace)

$$fp(w) = m - \frac{1}{n - w + 1} \left(\sum_{i=1}^m (f_i - w) I(f_i > w) \right. \\ \left. + \sum_{i=1}^m (l_i - w) I(l_i > w) \right. \\ \left. + n \sum_{t=w+1}^{n-1} (t - w) P(rt = t) \right)$$

$$fp(w) \approx m - \sum_{t=w+1}^{n-1} (t - w) P(rt = t)$$

**LRU miss rate
correctness condition
(Corollary 2.3)**

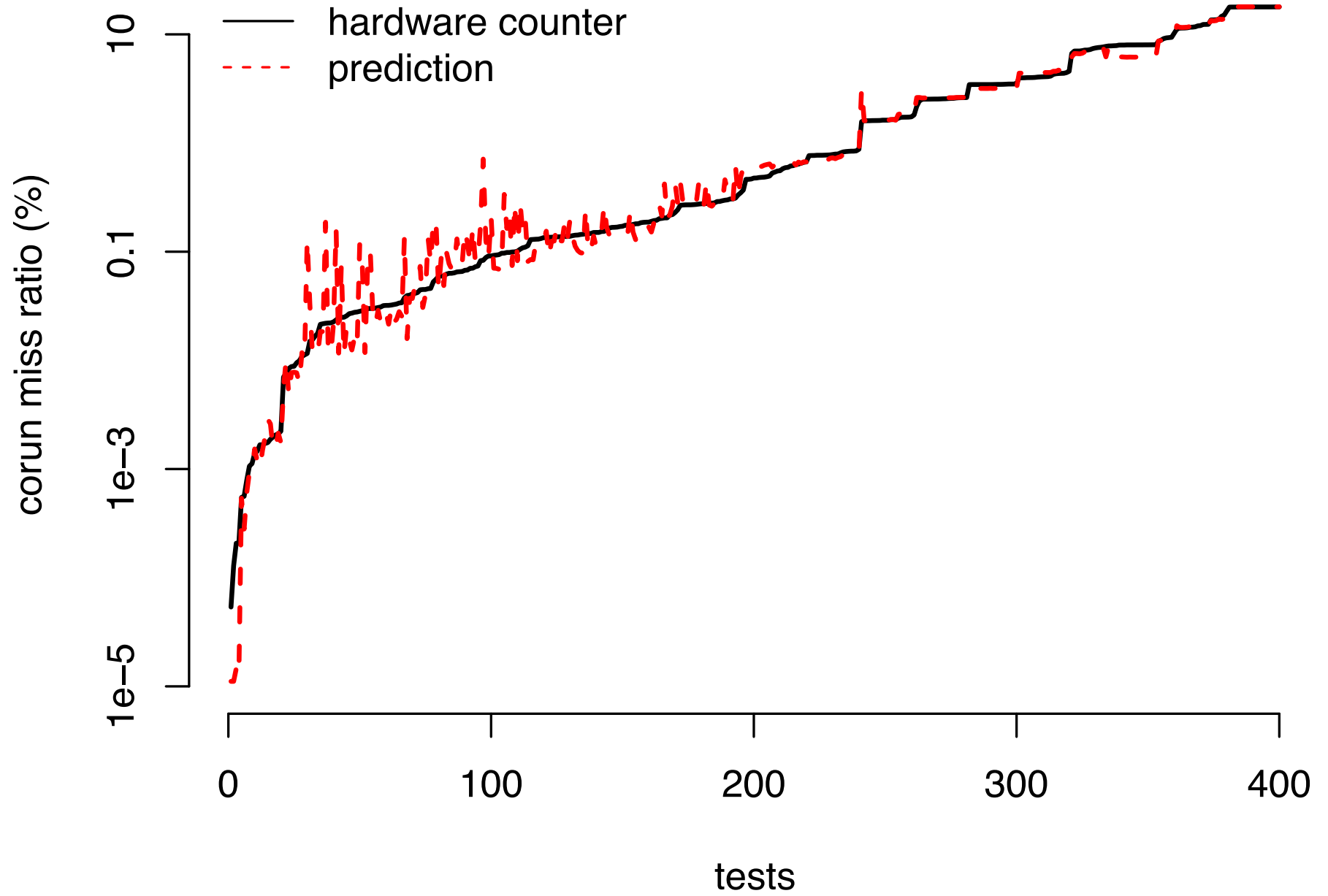
Denning's Law of Locality

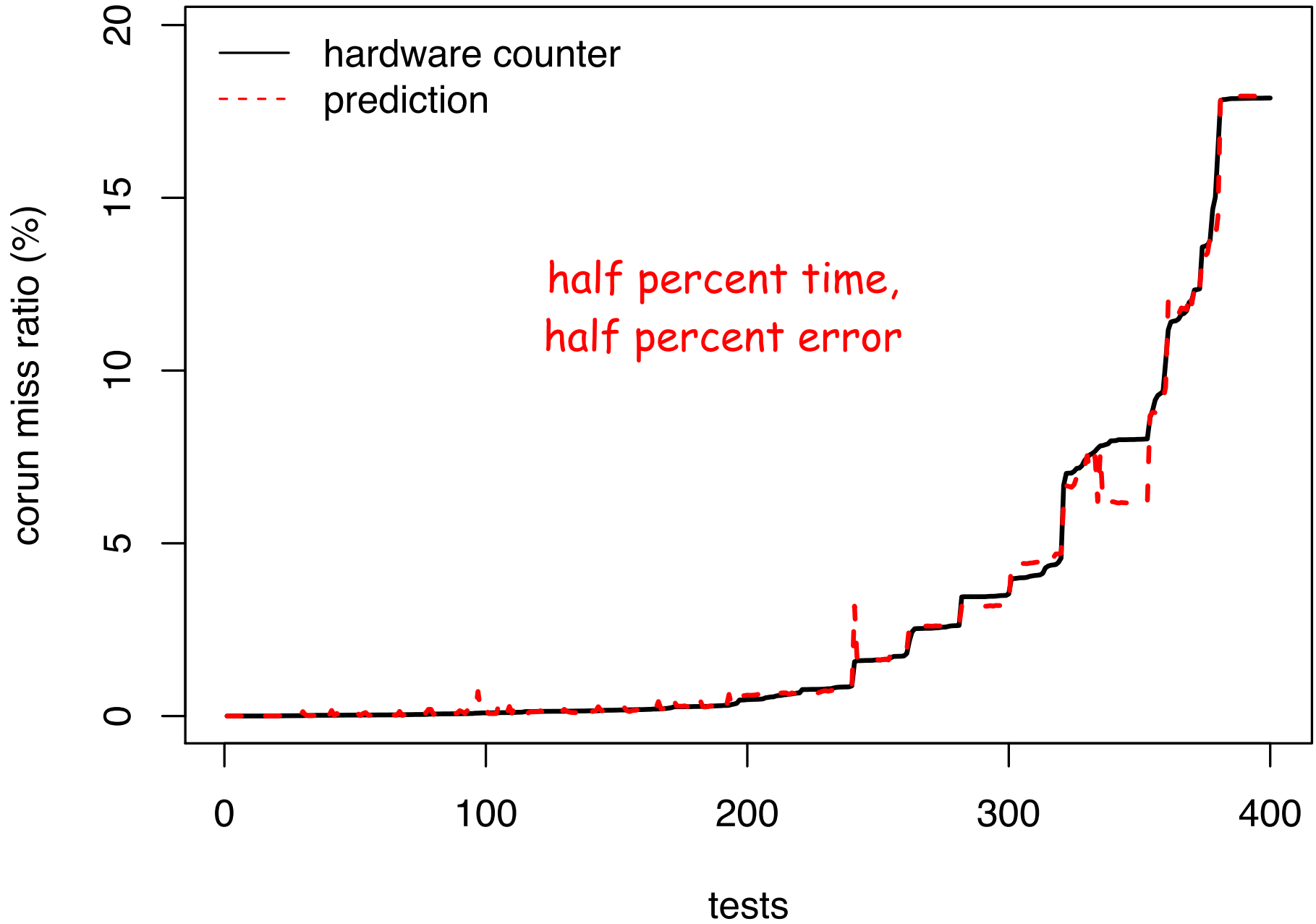
Reuse Window Hypothesis

The footprints in **all reuse windows** have the same distribution as the footprints in **all windows**

Modeling of Cache Sharing w/o Parallel Testing

- 20 SPEC 2006 programs
 - 190 different pair runs
- Modeling
 - per program footprint
 - composition
 - Xiang et al. (POPP'11/PACT'11) based on Chandra et al. (HPCA'05), Suh et al. (ICS'01), and Thiebaut & Stone (TOCS'87)
 - a few hours
 - prediction for all cache sizes
- Measurement
 - 190 pair runs
 - 380 hw counter reads (OFFCORE.DATA_IN, 8MB 16-way L3)
 - ~9 days total CPU time





Summary of the Theory

- Locality metrics
 - miss rate in hardware
 - reuse distance and footprint in a program
- The higher-order theory of locality
 - mutual conversion
 - may all be computed from reuse time
 - Denning's law, the Xiang formula
- Correctness
 - theoretical
 - reuse-window hypothesis
 - empirical
 - "half and half" in pair-run tests

Recent Developments

- Parallel reuse distance measurement
 - cluster [OSU, IPDPS 2012]
 - GPU [ICT and NCSU, IPDPS 2012]
 - sampling
 - footprint shadow sampling [this paper]
 - multicore reuse distance [Purdue, PACT 2010]
- Reuse distance in threaded code
 - multicore reuse distance [Purdue, PACT 2010]
 - CRD/PRD scaling [Maryland, ISCA 2013, to appear]
- Shared footprint [Rochester, WODA 2013]
- Static reuse distance analysis in Matlab [Indiana, ICS 2010]
- Static footprint analysis [Rochester, CGO 2013]