Footprint-based Locality Analysis

Xiaoya Xiang, Bin Bao, Chen Ding
University of Rochester

2011-11-10

Memory Performance

- On modern computer system, memory performance depends on the active data usage.
  - primary factor affecting the latency of memory operations and the demand for memory bandwidth.
  - data interference in shared cache environment

- Locality = Active data usage
  - reuse distance model: upto thousands of times slowdown
  - footprint model

Reuse Distance

- Definition
    - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
    - the distribution of all finite reuse distances
    - determines working set size and gives the miss rate of fully associative cache of all sizes
        - associativity effect [Smith 1976]

Reuse Distance

- Definition
  - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
  - the distribution of all finite reuse distances
  - determines working set size and gives the miss rate of fully associative cache of all sizes
    - associativity effect [Smith 1976]

a b c a a c b

Reuse Distance

- Definition
  - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
  - the distribution of all finite reuse distances
  - determines working set size and gives the miss rate of fully associative cache of all sizes
    - associativity effect [Smith 1976]

a b c a a c $\overset{2}{\text{b}}$

Reuse Distance

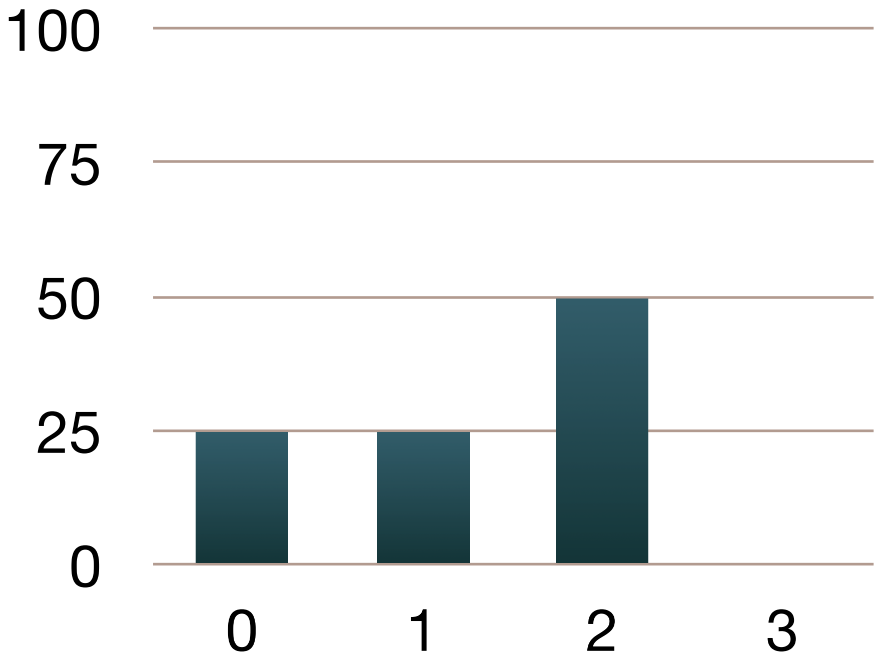
- Definition
  - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
  - the distribution of all finite reuse distances
  - determines working set size and gives the miss rate of fully associative cache of all sizes
    - associativity effect [Smith 1976]

∞	∞	∞	2	0	1	2
a	b	c	a	a	c	b

Reuse Distance

- Definition
  - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
  - the distribution of all finite reuse distances
  - determines working set size and gives the miss rate of fully associative cache of all sizes
    - associativity effect [Smith 1976]

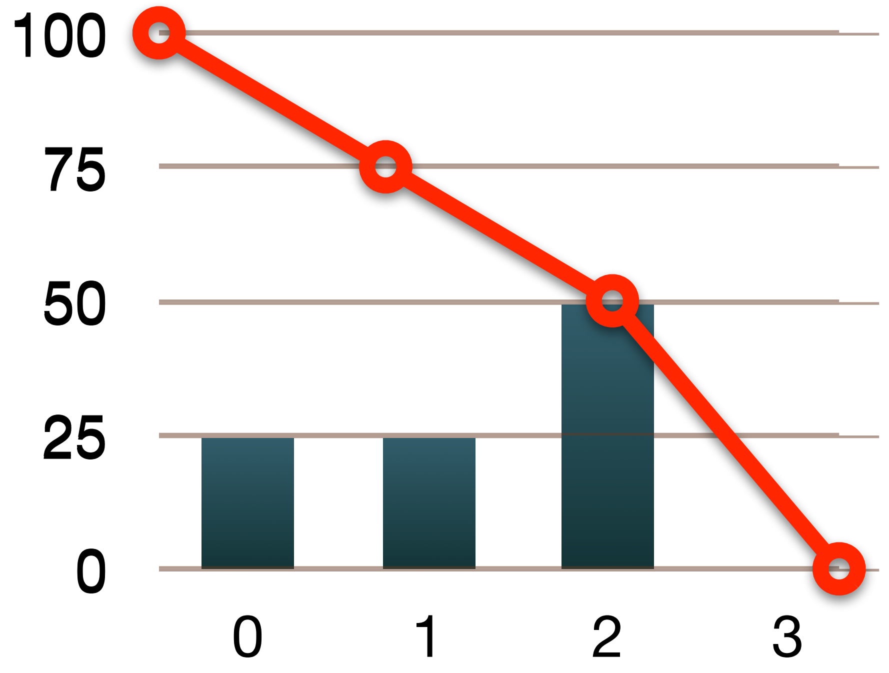
∞	∞	∞	2	0	1	2
a	b	c	a	a	c	b



Reuse Distance

- Definition
  - the number of distinct elements accessed between two consecutive accesses to the same data
- Reuse signature of an execution
  - the distribution of all finite reuse distances
  - determines working set size and gives the miss rate of fully associative cache of all sizes
    - associativity effect [Smith 1976]

∞ ∞ ∞ 2 0 1 2
a b c a a c b



Reuse Distance Measurement

Measurement algorithms since 1970	Time	Space
Naive counting	$O(N^2)$	O(N)
Trace as a stack [IBM’70]	O(NM)	O(M)
Trace as a vector [IBM’75, Illinois’02]	O(NlogN)	O(N)
Trace as a tree [LBNL’81], splay tree [Michigan’93], interval tree [Illinois’02]	O(NlogM)	O(M)
Fixed cache sizes [Winsconsin’91]	O(N)	O(C)
Approximation tree [Rochester’03]	O(NloglogM)	O(logM)
Approx. using time [Rochester’07]	O(N)	O(1)

N is the length of the trace. M is the size of data. C is the size of cache.

Reuse Distance Measurement

Measurement algorithms since 1970	Time	Space
Naive counting	$O(N^2)$	$O(N)$
Trace as a stack [IBM’70]	$O(NM)$	$O(M)$
Trace as a vector [IBM’75, Illinois’02]	$O(N\log N)$	$O(N)$
Trace as a tree [LBNL’81], splay tree [Michigan’93], interval tree [Illinois’02]	$O(N\log M)$	$O(M)$
Fixed cache sizes [Winsconsin’91]	$O(N)$	$O(C)$
Approximation tree [Rochester’03]	$O(N\log\log M)$	$O(\log M)$
Approx. using time [Rochester’07]	$O(N)$	$O(1)$

N is the length of the trace. M is the size of data. C is the size of cache.

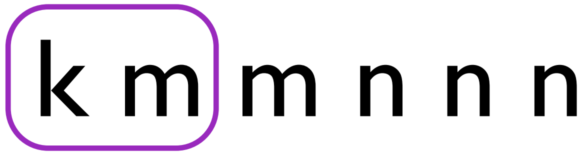
Footprint

- Definition
  - given an execution window in a trace, the footprint is the number of distinct elements accessed in the window

k m m n n n

Footprint

- Definition
  - given an execution window in a trace, the footprint is the number of distinct elements accessed in the window

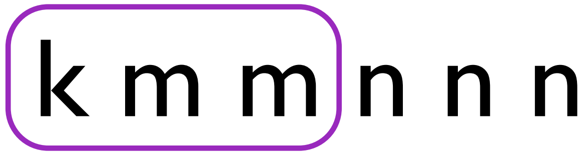


k m m n n n

window size= 2 footprint=2

Footprint

- Definition
  - given an execution window in a trace, the footprint is the number of distinct elements accessed in the window

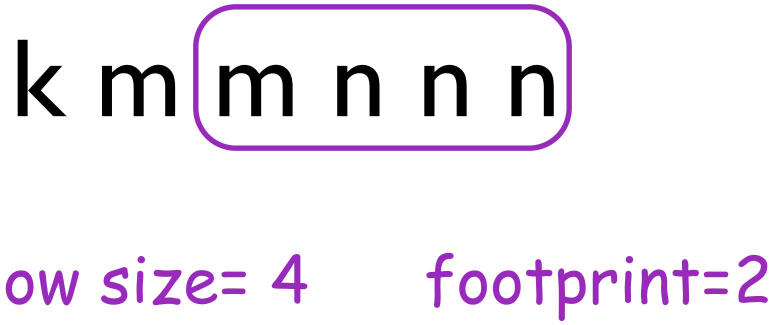


k m m n n n

window size= 3   footprint=2

Footprint

- Definition
  - given an execution window in a trace, the footprint is the number of distinct elements accessed in the window

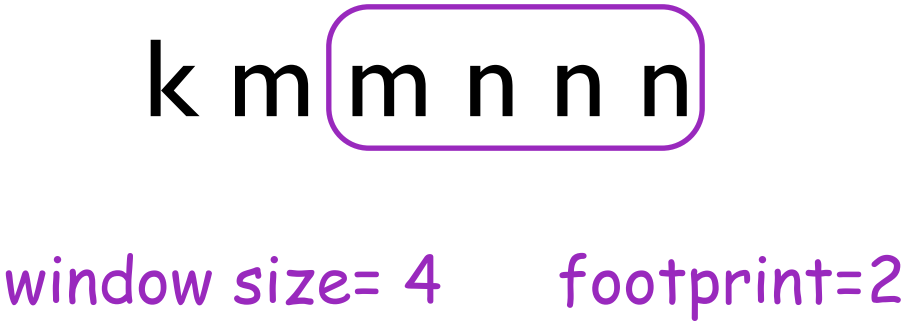


k m m n n n

window size= 4 footprint=2

Footprint

- Definition
  - given an execution window in a trace, the footprint is the number of distinct elements accessed in the window



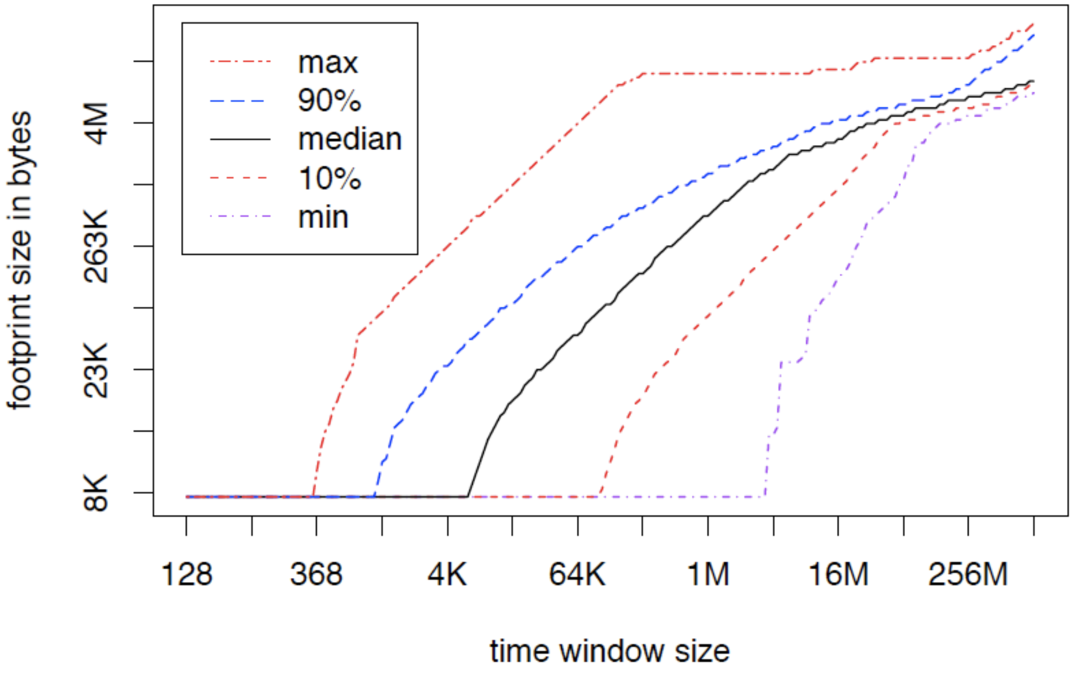
k m m n n n

window size= 4 footprint=2

- All-Footprint statistic
  - a distribution of footprint size over window size
  - precise distribution requires measuring all windows: N(N+1)/2 windows in a N-long trace
- Another Model of Active Data Usage
  - a harder problem (than reuse distance)

All-footprint CKlogM Alg. [Xiang+ PPoPP'11]

- The algorithm
  - footprint counting
  - relative precision approximation
  - trace compression



- Efficiency
  - it is the first algorithm which can make complete measurement of all-footprint.
  - the cost is still too high for real-size workloads.
- Solution
  - confining to the average rather than the full range.

Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size t, average footprint takes average over all windows of length t.
- Example

a b b b

when window size equals 2

footprint =

Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size `t`, average footprint takes average over all windows of length `t`.
- Example

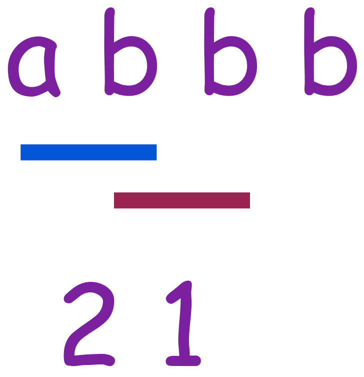
$\underline{a\ b}\ b\ b$

when window size equals 2

footprint = 2

Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size `t`, average footprint takes average over all windows of length `t`.
- Example

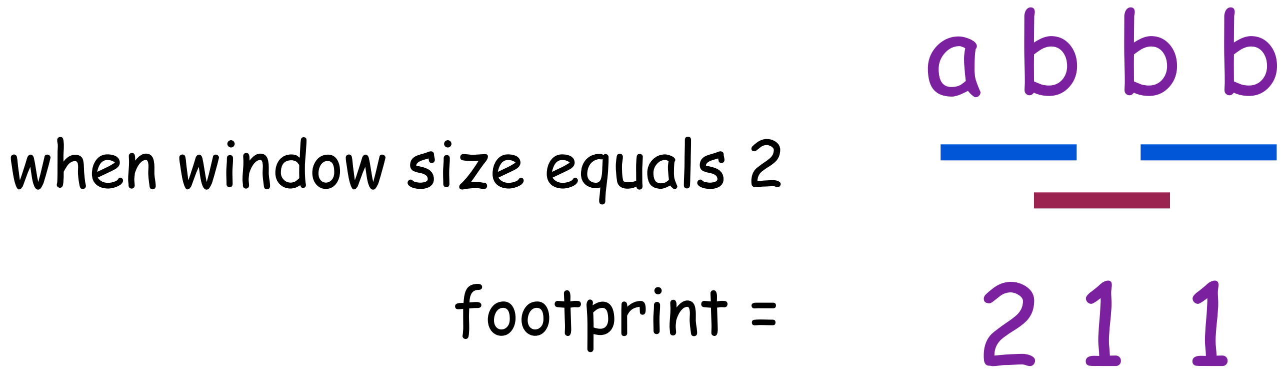


when window size equals 2

footprint =

Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size t, average footprint takes average over all windows of length t.
- Example



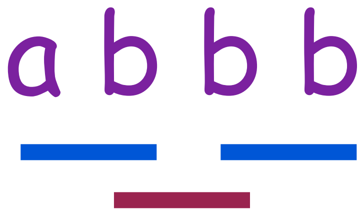
a b b b

when window size equals 2

footprint = 2 1 1

Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size `t`, average footprint takes average over all windows of length `t`.
- Example



when window size equals 2

a b b b

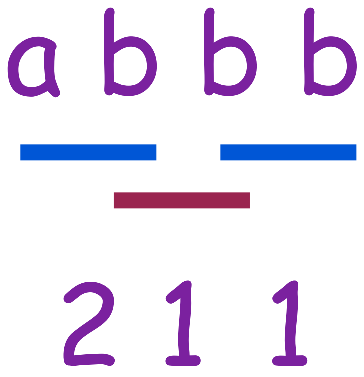
footprint = 2 1 1

$$\overline{fp}(2) = (2+1+1)/3 = 4/3$$

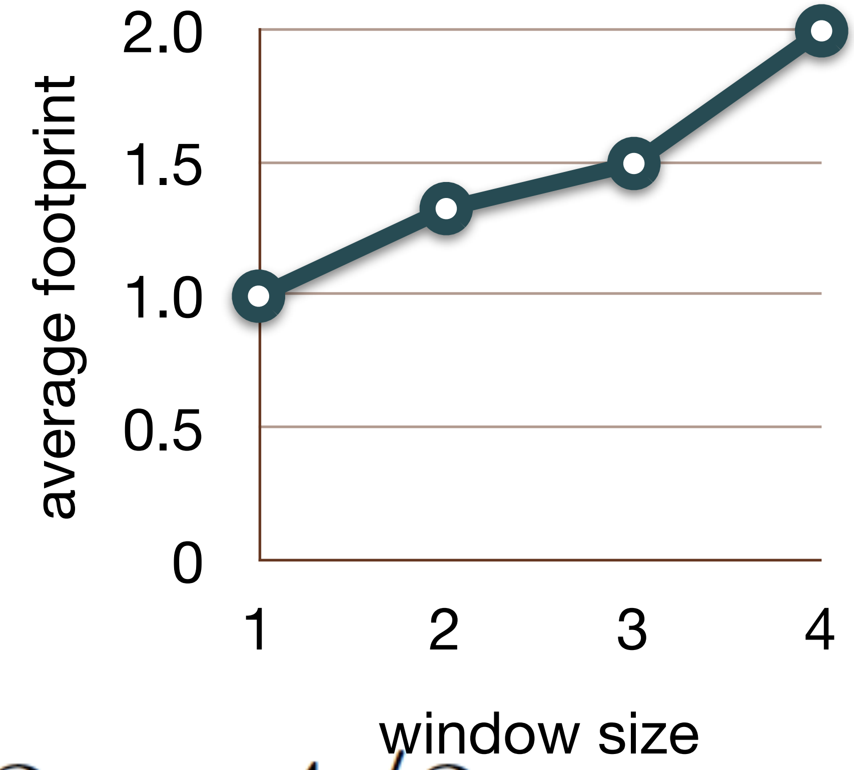
Average Footprint O(N) Algo. [Xiang+ PACT'11]

- Given a trace and a window size t, average footprint takes average over all windows of length t.
- Example

when window size equals 2

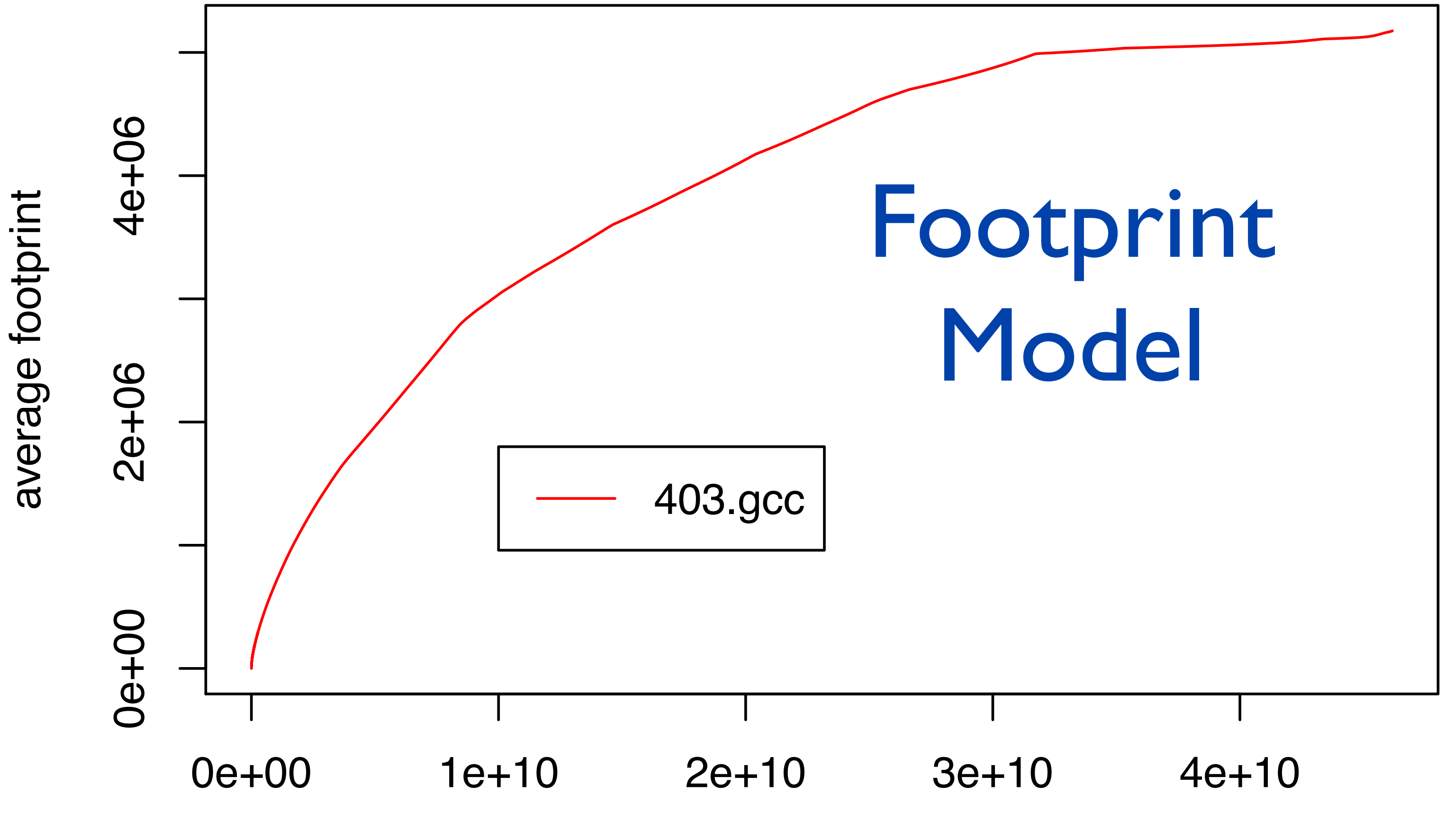


footprint = 2 1 1

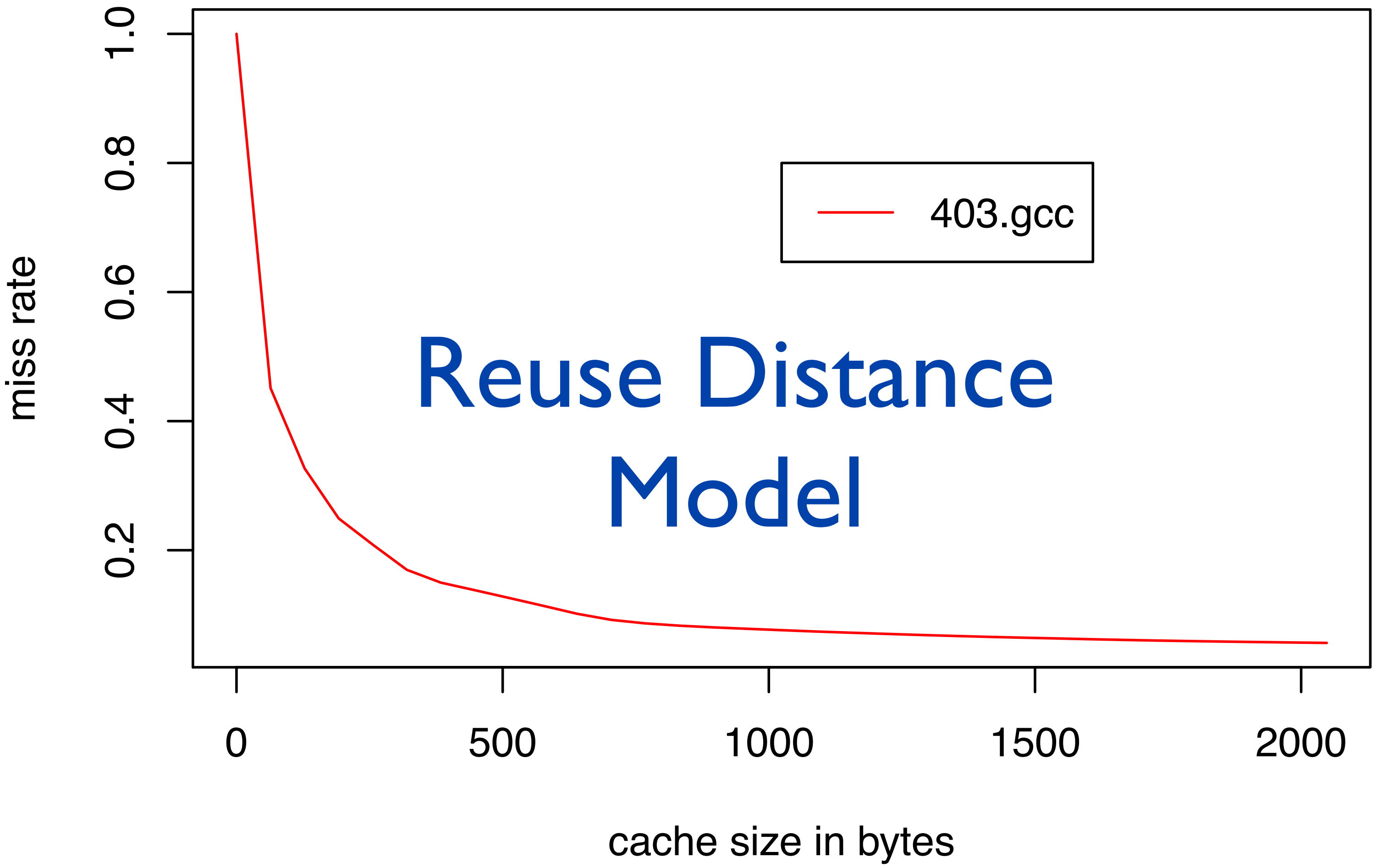


$$\overline{fp}(2) = (2+1+1)/3 = 4/3$$

- Compared to hardware counters
  - all cache sizes, no perturbation (deterministic results)



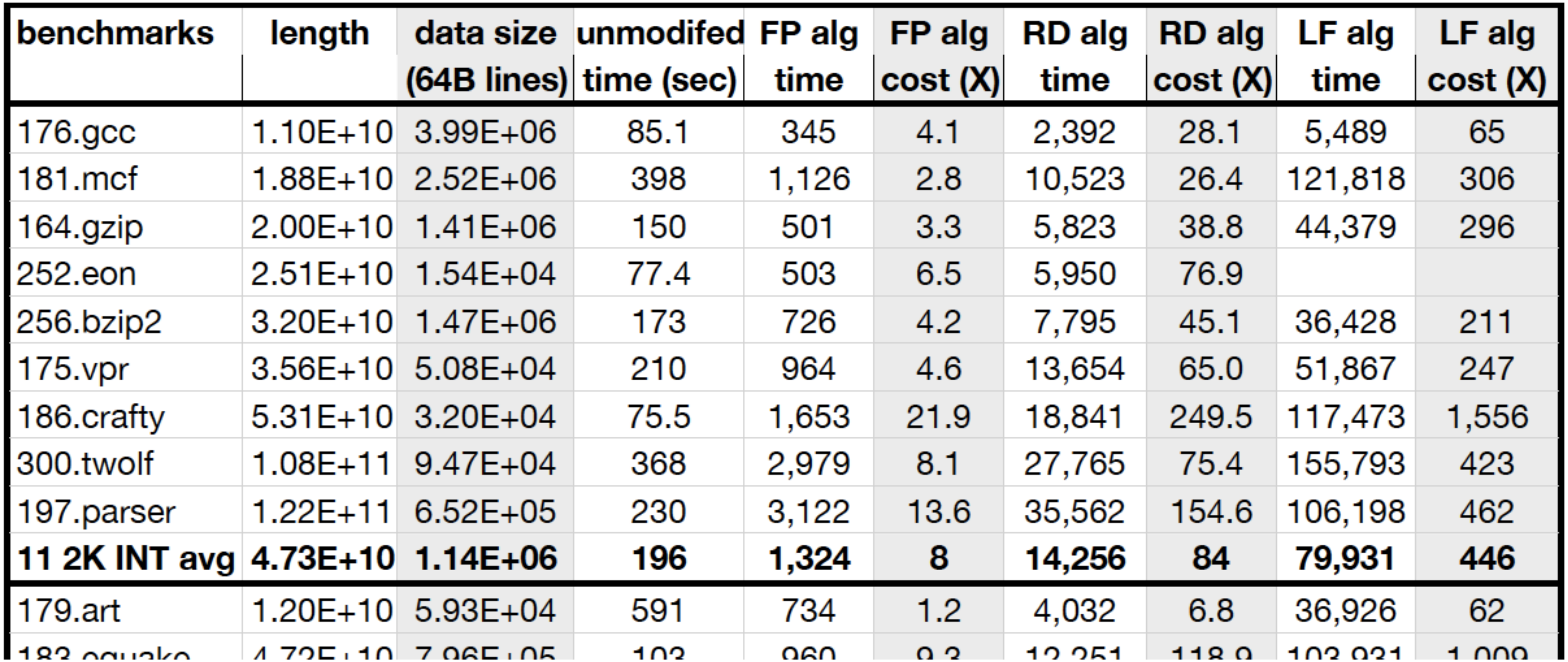
Footprint Model



Reuse Distance Model

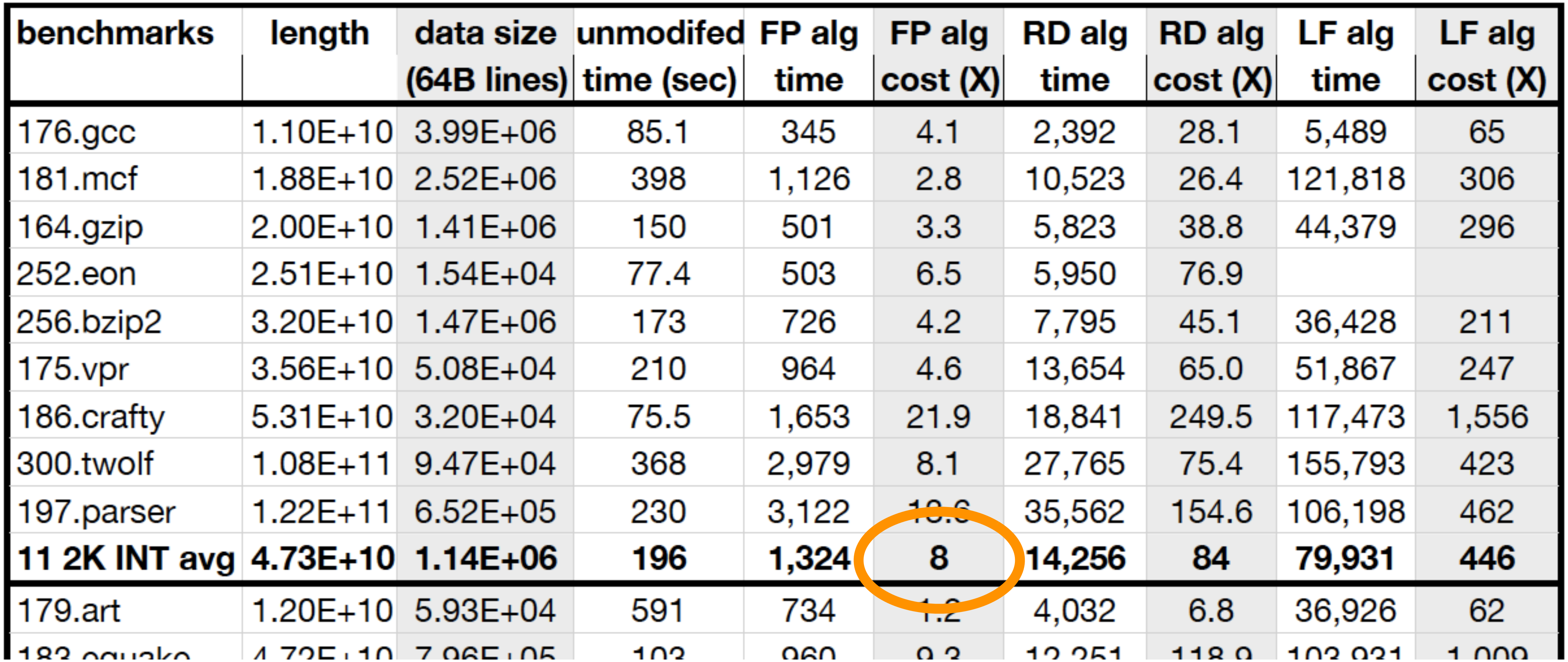
- Compared to reuse distance
  - direct time/space relation, more intuitive
  - O(n) vs. O(nloglogm)
  - relation to miss rate?

Footprint Analysis is Faster [PACT 11]



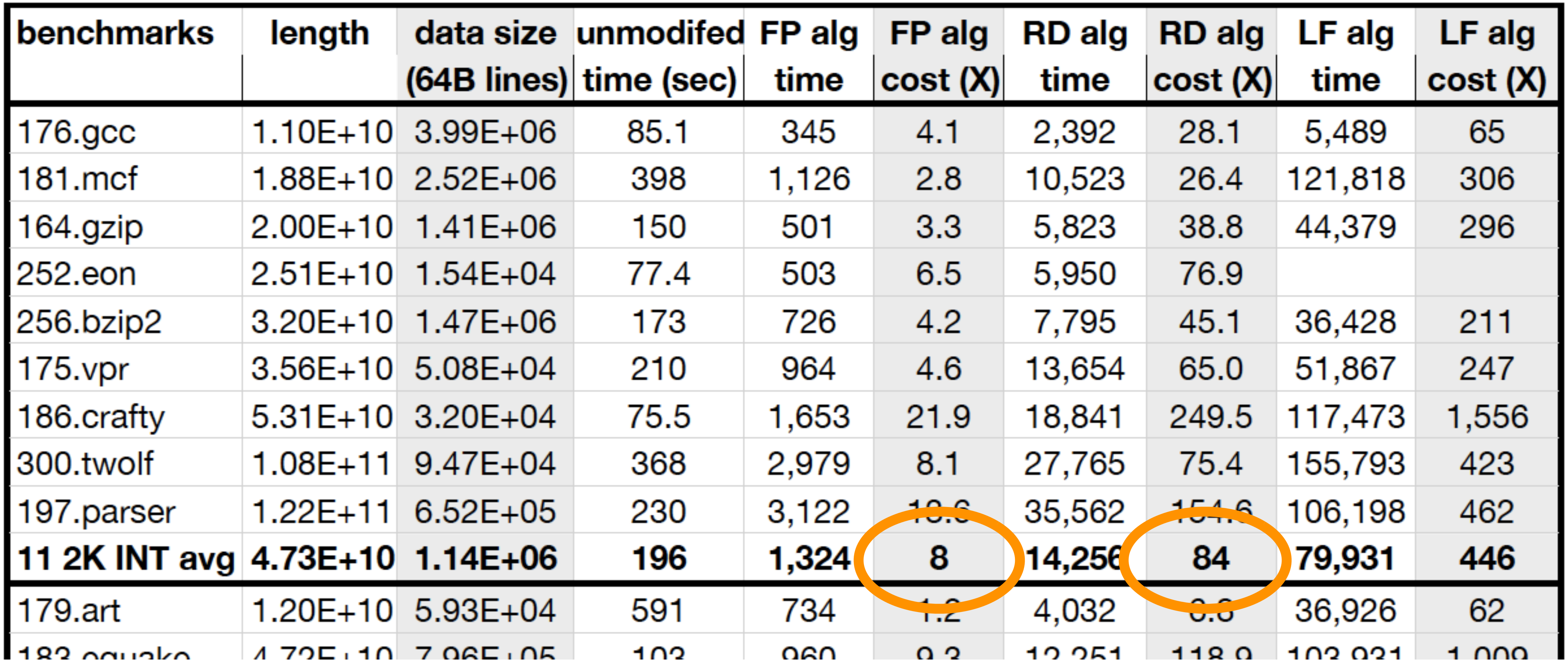
benchmarks	length	data size (64B lines)	unmodifed time (sec)	FP alg time	FP alg cost (X)	RD alg time	RD alg cost (X)	LF alg time	LF alg cost (X)
176.gcc	1.10E+10	3.99E+06	85.1	345	4.1	2,392	28.1	5,489	65
181.mcf	1.88E+10	2.52E+06	398	1,126	2.8	10,523	26.4	121,818	306
164.gzip	2.00E+10	1.41E+06	150	501	3.3	5,823	38.8	44,379	296
252.eon	2.51E+10	1.54E+04	77.4	503	6.5	5,950	76.9		
256.bzip2	3.20E+10	1.47E+06	173	726	4.2	7,795	45.1	36,428	211
175.vpr	3.56E+10	5.08E+04	210	964	4.6	13,654	65.0	51,867	247
186.crafty	5.31E+10	3.20E+04	75.5	1,653	21.9	18,841	249.5	117,473	1,556
300.twolf	1.08E+11	9.47E+04	368	2,979	8.1	27,765	75.4	155,793	423
197.parser	1.22E+11	6.52E+05	230	3,122	13.6	35,562	154.6	106,198	462
11 2K INT avg	**4.73E+10**	**1.14E+06**	**196**	**1,324**	**8**	**14,256**	**84**	**79,931**	**446**
179.art	1.20E+10	5.93E+04	591	734	1.2	4,032	6.8	36,926	62
183.equake	4.72E+10	7.96E+05	103	960	9.3	12,251	118.9	103,931	1,009

Footprint Analysis is Faster [PACT 11]



benchmarks	length	data size (64B lines)	unmodifed time (sec)	FP alg time	FP alg cost (X)	RD alg time	RD alg cost (X)	LF alg time	LF alg cost (X)
176.gcc	1.10E+10	3.99E+06	85.1	345	4.1	2,392	28.1	5,489	65
181.mcf	1.88E+10	2.52E+06	398	1,126	2.8	10,523	26.4	121,818	306
164.gzip	2.00E+10	1.41E+06	150	501	3.3	5,823	38.8	44,379	296
252.eon	2.51E+10	1.54E+04	77.4	503	6.5	5,950	76.9		
256.bzip2	3.20E+10	1.47E+06	173	726	4.2	7,795	45.1	36,428	211
175.vpr	3.56E+10	5.08E+04	210	964	4.6	13,654	65.0	51,867	247
186.crafty	5.31E+10	3.20E+04	75.5	1,653	21.9	18,841	249.5	117,473	1,556
300.twolf	1.08E+11	9.47E+04	368	2,979	8.1	27,765	75.4	155,793	423
197.parser	1.22E+11	6.52E+05	230	3,122	13.6	35,562	154.6	106,198	462
11 2K INT avg	**4.73E+10**	**1.14E+06**	**196**	**1,324**	**8**	**14,256**	**84**	**79,931**	**446**
179.art	1.20E+10	5.93E+04	591	734	1.2	4,032	6.8	36,926	62
183.equake	4.72E+10	7.96E+05	103	960	9.3	12,251	118.9	103,931	1,009

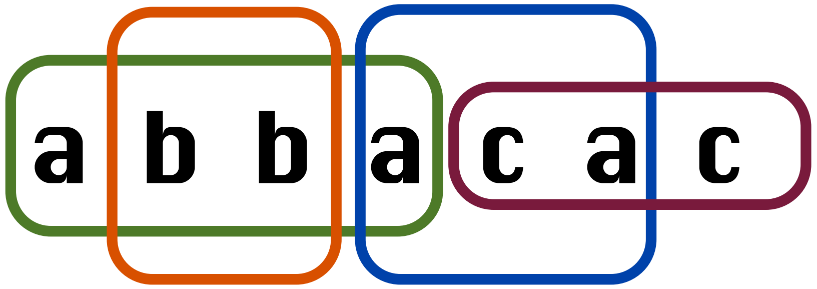
Footprint Analysis is Faster [PACT 11]



benchmarks	length	data size (64B lines)	unmodifed time (sec)	FP alg time	FP alg cost (X)	RD alg time	RD alg cost (X)	LF alg time	LF alg cost (X)
176.gcc	1.10E+10	3.99E+06	85.1	345	4.1	2,392	28.1	5,489	65
181.mcf	1.88E+10	2.52E+06	398	1,126	2.8	10,523	26.4	121,818	306
164.gzip	2.00E+10	1.41E+06	150	501	3.3	5,823	38.8	44,379	296
252.eon	2.51E+10	1.54E+04	77.4	503	6.5	5,950	76.9		
256.bzip2	3.20E+10	1.47E+06	173	726	4.2	7,795	45.1	36,428	211
175.vpr	3.56E+10	5.08E+04	210	964	4.6	13,654	65.0	51,867	247
186.crafty	5.31E+10	3.20E+04	75.5	1,653	21.9	18,841	249.5	117,473	1,556
300.twolf	1.08E+11	9.47E+04	368	2,979	8.1	27,765	75.4	155,793	423
197.parser	1.22E+11	6.52E+05	230	3,122	10.6	35,562	154.6	106,198	462
11 2K INT avg	**4.73E+10**	**1.14E+06**	**196**	**1,324**	**8**	**14,256**	**84**	**79,931**	**446**
179.art	1.20E+10	5.93E+04	591	734	1.2	4,032	6.8	36,926	62
183.equake	4.72E+10	7.96E+05	103	960	9.3	12,251	118.9	103,931	1,009

Footprint to Reuse Distance Conversion

- Use the average footprint in all windows as the average for all reuse windows
- An example trace:

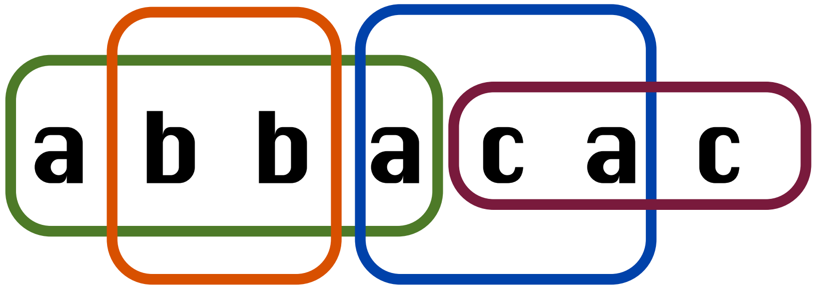


rd	2	1	2	2
reuse ws:w	4	2	3	3
avg. fp(w)	2.5	1.83	2.2	2.2
approx. rd	2.5	1.83	2.2	2.2

- Footprints can be easily sampled

Footprint to Reuse Distance Conversion

- Use the average footprint in all windows as the average for all reuse windows
- An example trace:

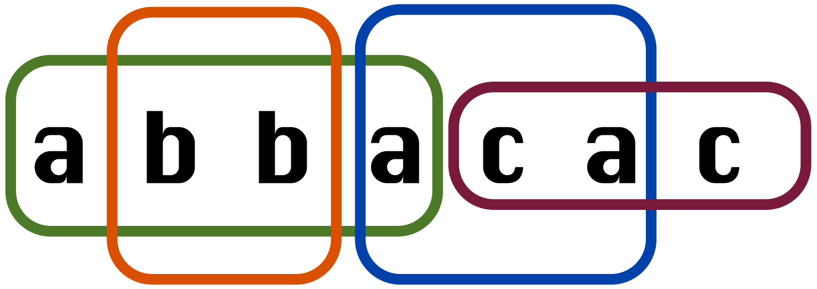


rd	2	1	2	2
reuse ws:w	4	2	3	3
avg. fp(w)	2.5	1.83	2.2	2.2
approx. rd	2.5	1.83	2.2	2.2

- Footprints can be easily sampled

Footprint to Reuse Distance Conversion

- Use the average footprint in all windows as the average for all reuse windows
- An example trace:



rd	2	1	2	2
reuse ws:w	4	2	3	3
avg. fp(w)	2.5	1.83	2.2	2.2
approx. rd	2.5	1.83	2.2	2.2

- Footprints can be easily sampled

Footprint Sampling

- footprint by definition is amenable to sampling since footprint window has known boundaries.
- disjoint footprint windows can be measured completely in parallel.
- shadow profiling

Evaluation: Analysis Speed

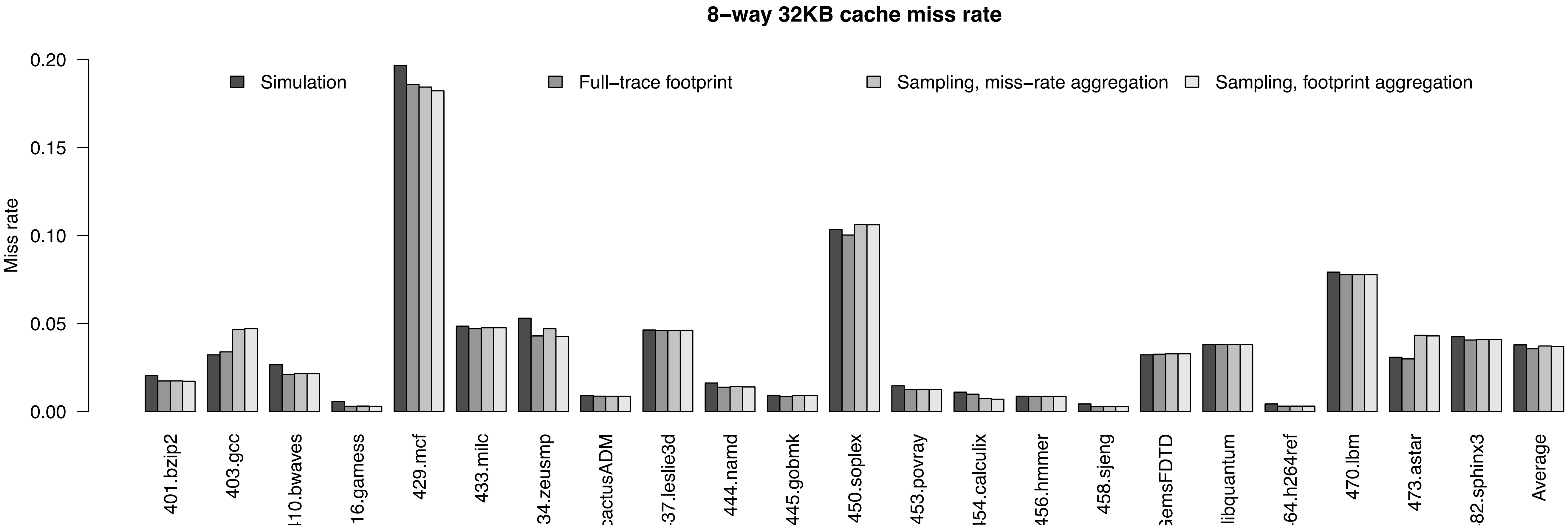
- Experimental Setup
  - full set of SPEC2006
  - instrument by Pin
  - profile on a Linux cluster

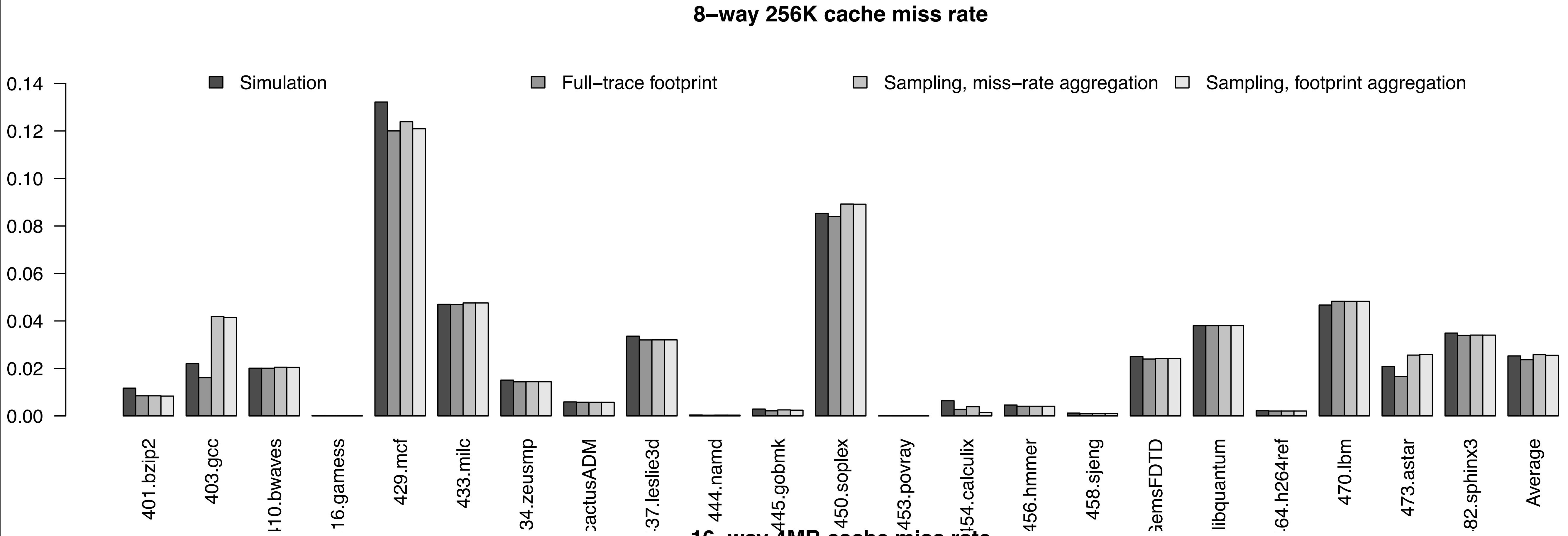
- Analysis Speed

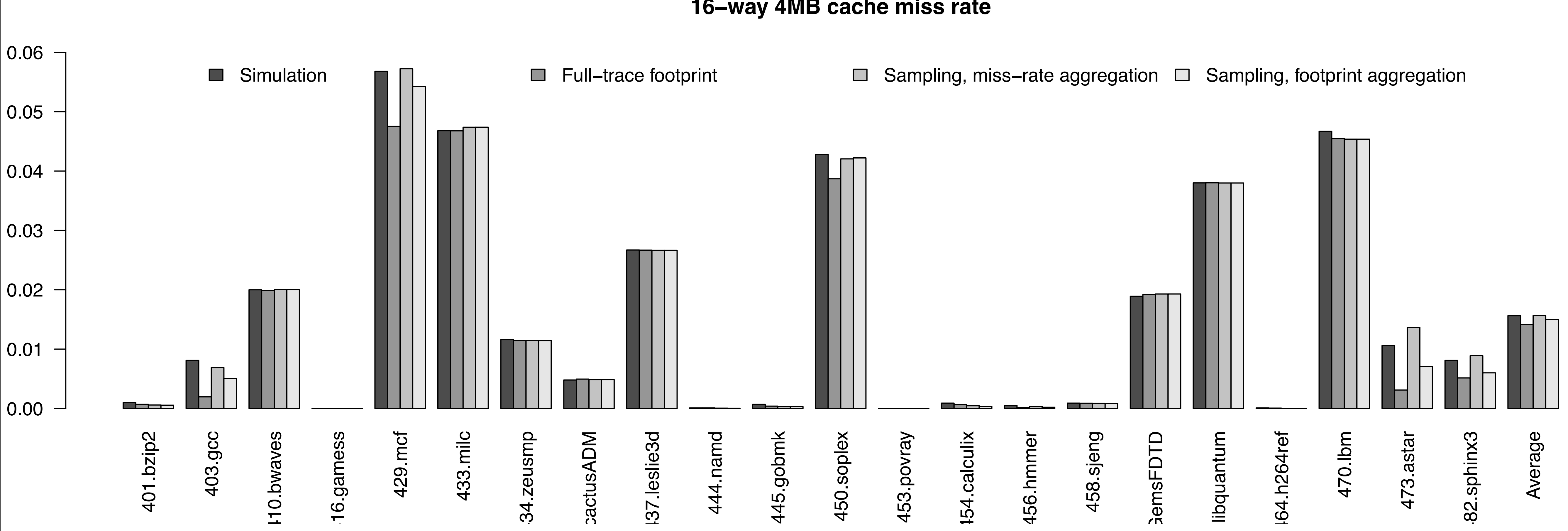
	orig (sec)	rd slowdown	fp slowdown	fp-sampling slowdown
max	1302.82 (436.cactus)	688x (456.hmmer)	40x (464.h264ref)	47% (416.gamess)
min	30.57 (403.gcc)	104x (429.mcf)	10x (429.mcf)	6% (456.hmmer)
mean	434.1	300x	21x	17%

Evaluation: Accuracy of Miss Rate Prediction

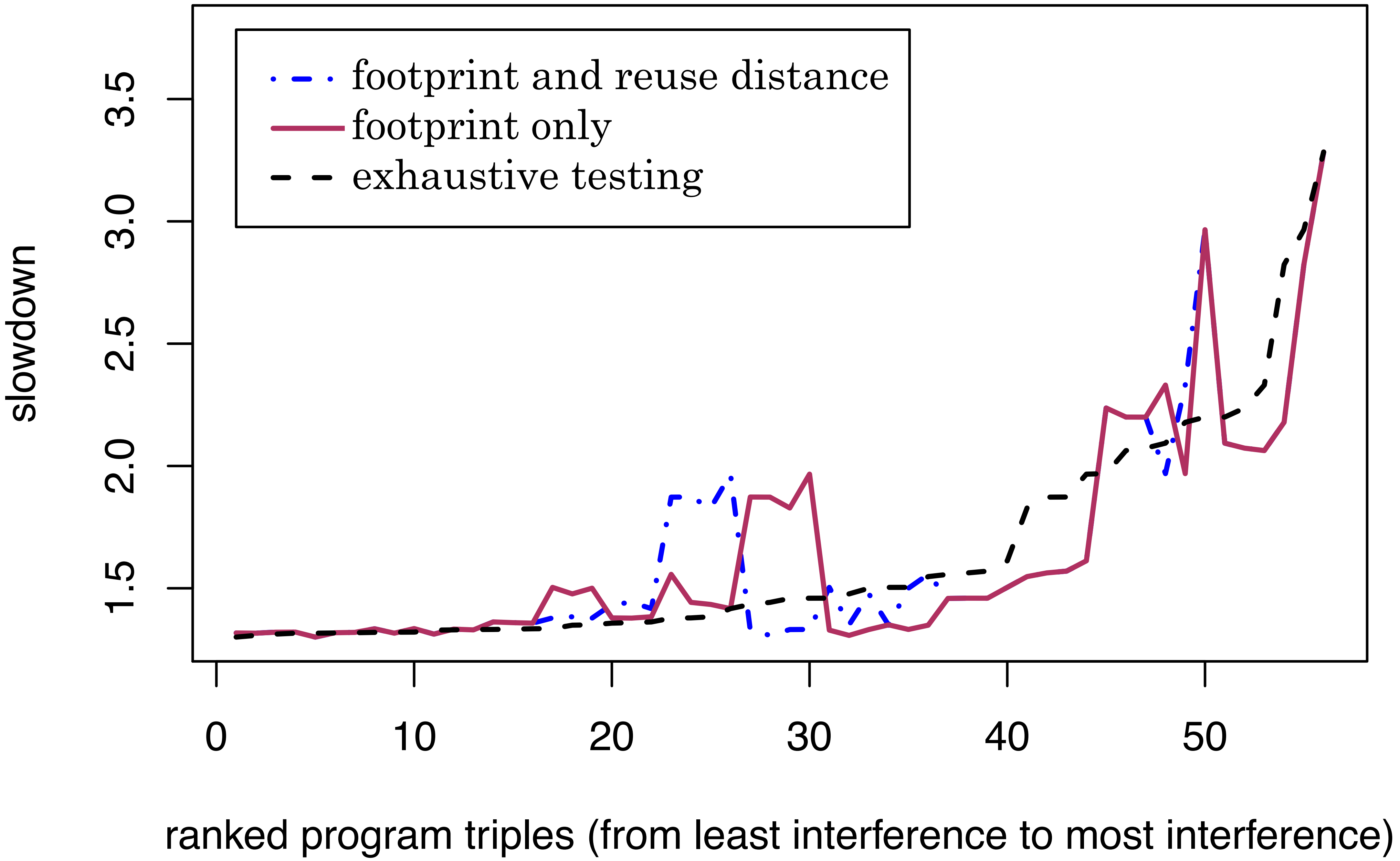
- use Smith equation [ICSE'76] to compute effect of associativity
- compare with 3-level cache simulations
  - 32KB, 8-way L1 data cache
  - 256KB, 8-way L2 cache
  - 4MB, 16-way L3 cache




8–way 256K cache miss rate
Simulation
Full–trace footprint
Sampling, miss–rate aggregation
Sampling, footprint aggregation
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
401.bzip2
403.gcc
10.bwaves
16.gamess
429.mcf
433.milc
34.zeusmp
cactusADM
37.leslie3d
444.namd
445.gobmk
450.soplex
453.povray
54.calculix
456.hmmer
458.sjeng
emsFDTD
libquantum
64.h264ref
470.lbm
473.astar
82.sphinx3
Average


16–way 4MB cache miss rate
Simulation
Full–trace footprint
Sampling, miss–rate aggregation
Sampling, footprint aggregation
0.06
0.05
0.04
0.03
0.02
0.01
0.00
401.bzip2
403.gcc
10.bwaves
16.gamess
429.mcf
433.milc
34.zeusmp
cactusADM
37.leslie3d
444.namd
445.gobmk
450.soplex
453.povray
54.calculix
456.hmmer
458.sjeng
emsFDTD
libquantum
64.h264ref
470.lbm
473.astar
82.sphinx3
Average

Evaluation: Corun Slowdown Prediction



Summary

- Two contributions
  - establish the relation between the new footprint statistics and the traditional locality statistics.
  - enable accurate on-line locality and cache sharing analysis through parallel sampling at a marginal cost, on average 17% for SPEC2006 benchmarks.

- Thanks
- Q&A