Efficient MRC Construction with SHARDS

Carl Waldspurger   Nohhyun Park
Alexander Garthwaite   Irfan Ahmad

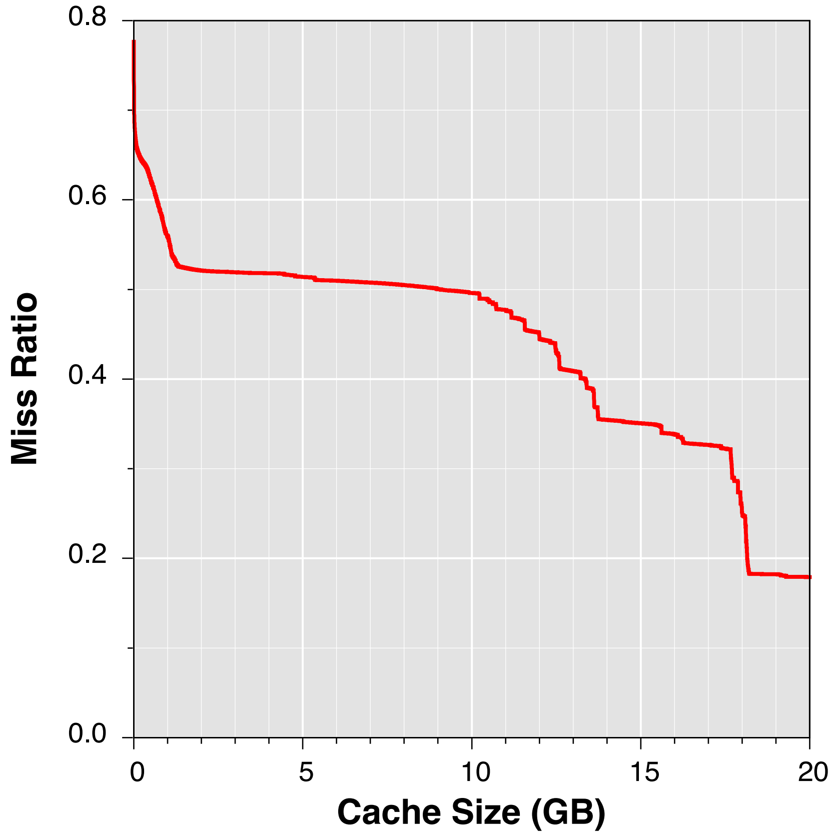
CloudPhysics, Inc.

USENIX Conference on File and Storage Technologies
February 17, 2015

Motivation

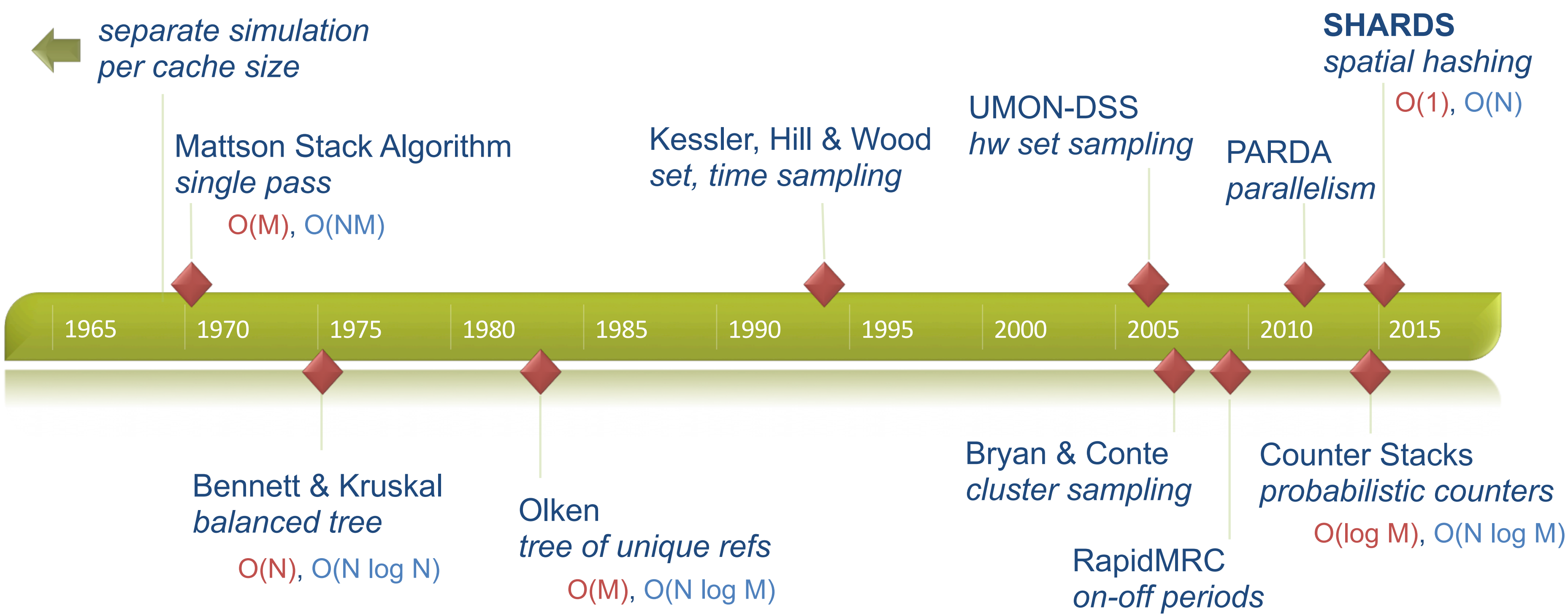
- Cache performance highly non-linear
- Benefit varies widely by workload
- Opportunity: dynamic cache management
  - Efficient sizing, allocation, and scheduling
  - Improve performance, isolation, QoS
- Problem: online modeling expensive
  - Too resource-intensive to be broadly practical
  - Exacerbated by increasing cache sizes

Modeling Cache Performance



- Miss Ratio Curve (MRC)
  - Performance as *f* (size)
  - Working set knees
  - Inform allocation policy
- Reuse distance
  - Unique intervening blocks between use and reuse
  - LRU, stack algorithms

MRC Algorithm Research

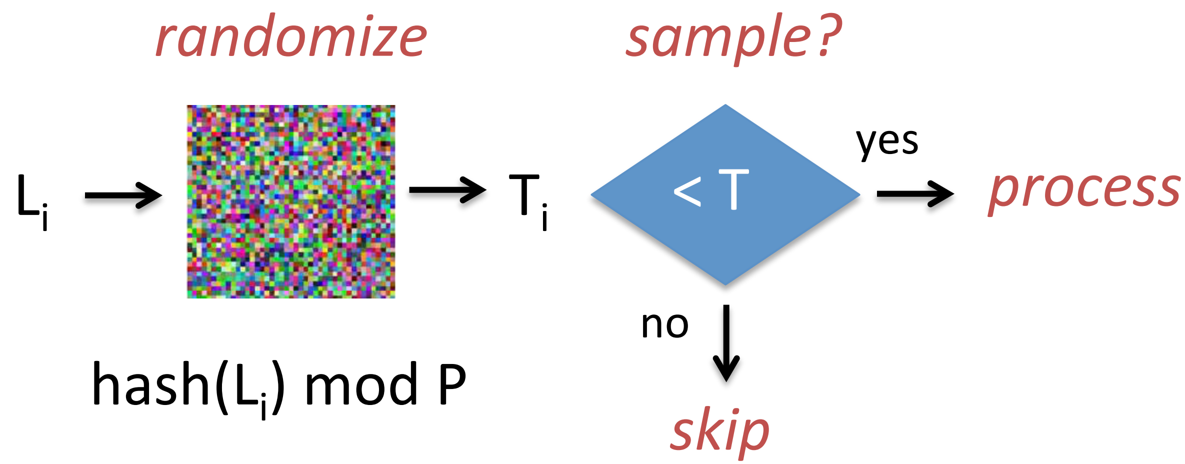


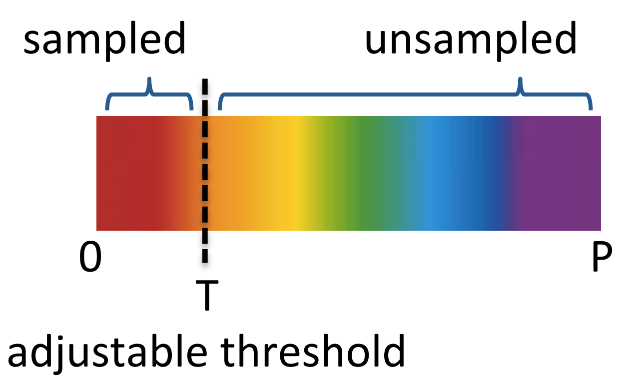
Space, Time Complexity
N = total refs, M = unique refs

Key Idea

- Track only a *small subset* of blocks
  - Filter input to existing algorithm
  - Run *full* algorithm, using only sampled blocks
  - Cheap/accurate enough for practical online MRCs?
- SHARDS approximation algorithm
  - Randomized spatial sampling
  - Uses hashing to capture all reuses of same block
  - High performance in tiny constant footprint
  - Surprisingly accurate MRCs

Spatially Hashed Sampling

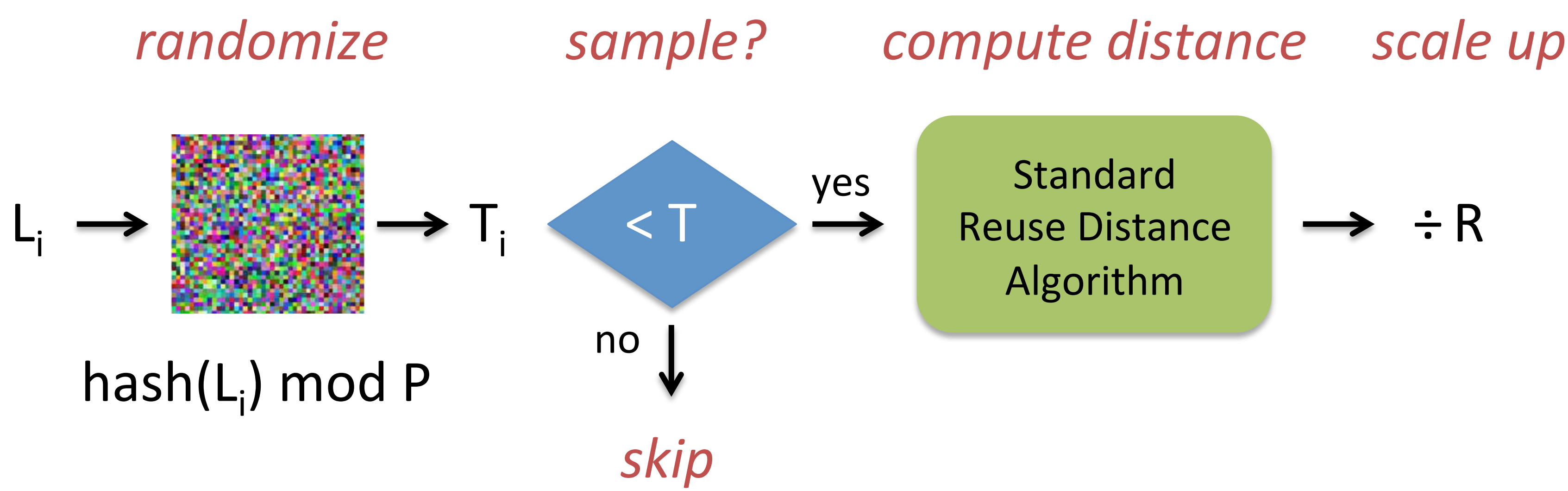




sampling rate R = T / P

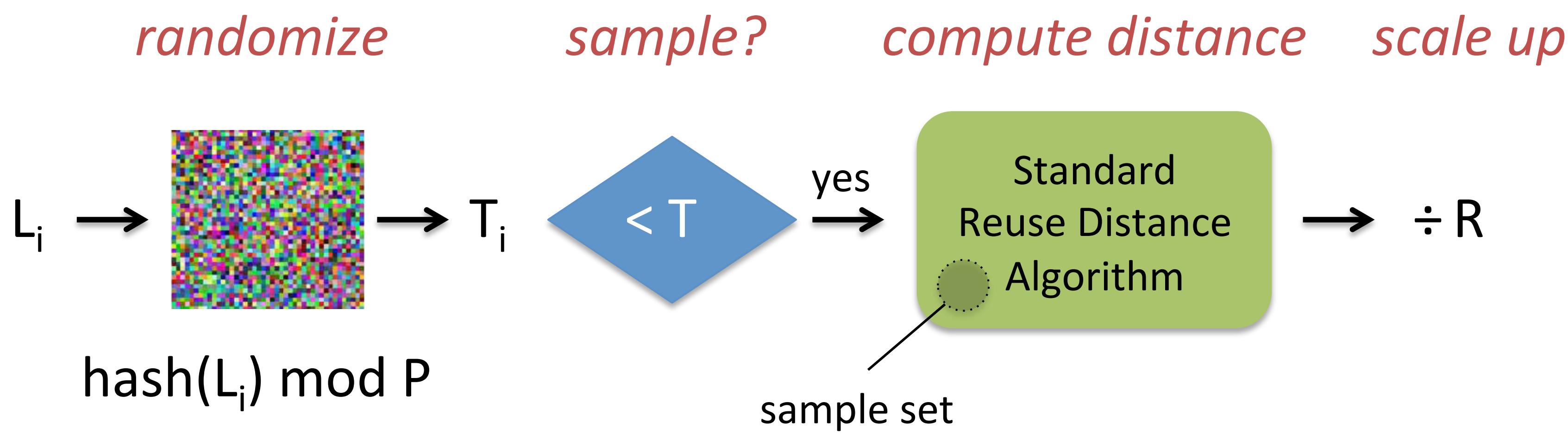
subset inclusion property
maintained as R is lowered

Basic SHARDS

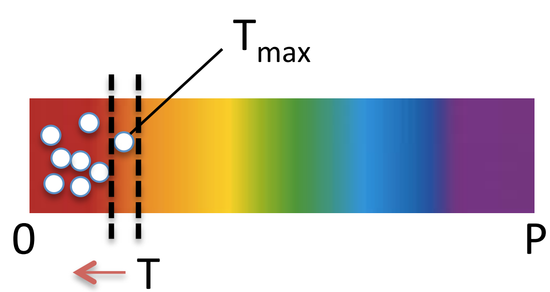


Each sample statistically represents $1/R$ blocks
Scale up reuse distances by same factor

SHARDS in Constant Space

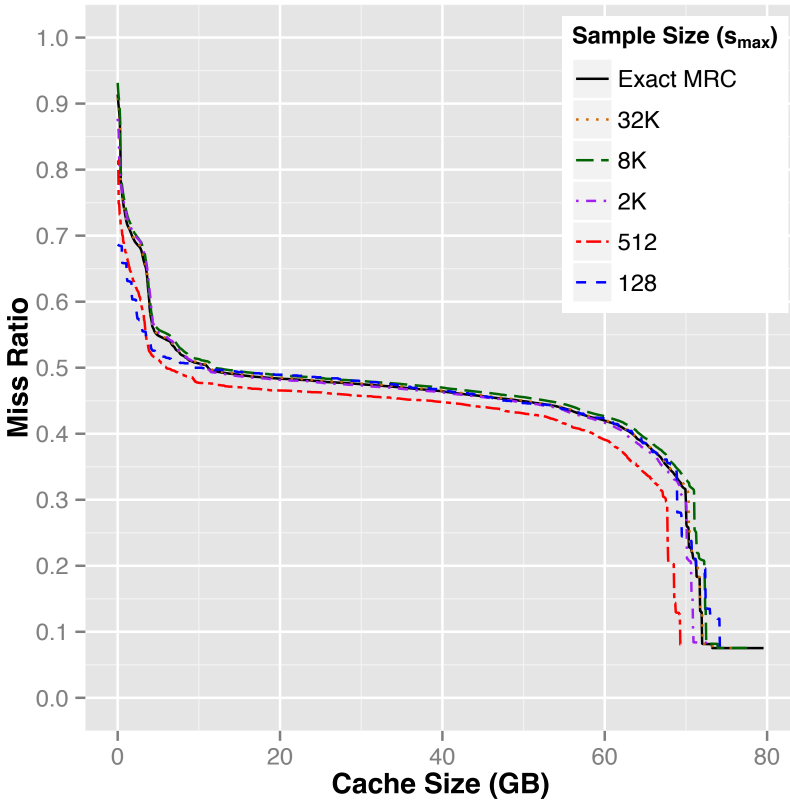


evict samples to bound set size



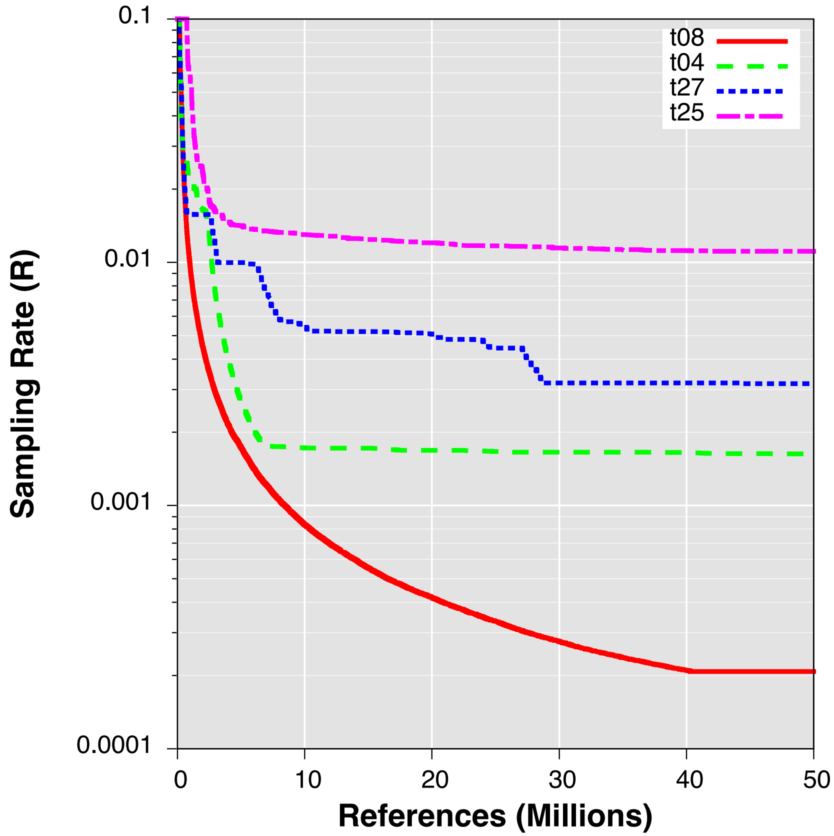
lower threshold $T = T_{max}$
reduces rate $R = T / P$

Example SHARDS MRCs



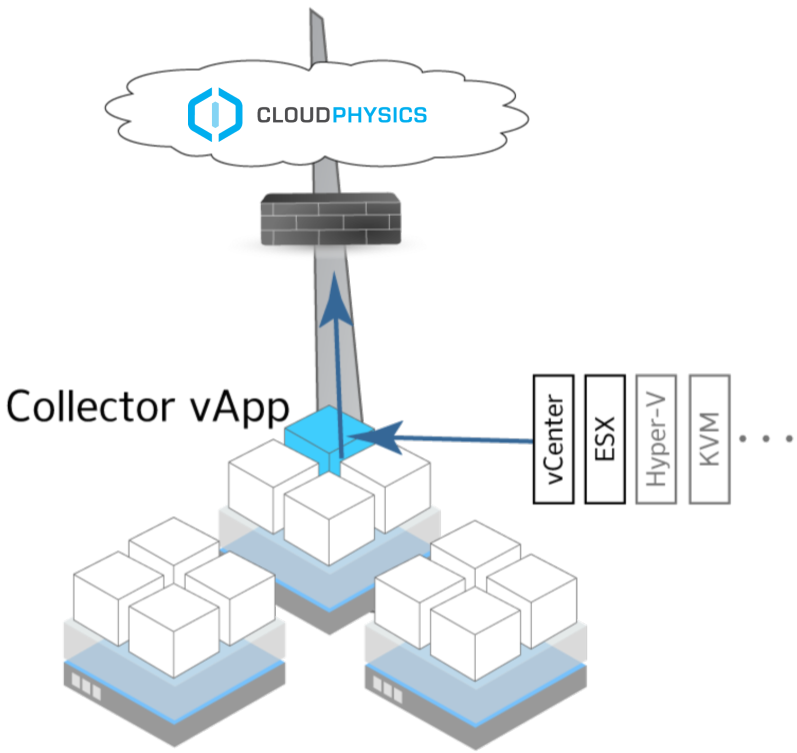
- Block I/O trace *t04*
  - Production VM disk
  - 69.5M refs, 5.2M unique
- Sample size s_{max}
  - Vary from 128 to 32K
  - $s_{max} \geq$ 2K very accurate
- Small constant footprint
- SHARDS_{adj} adjustment

Dynamic Rate Adaptation



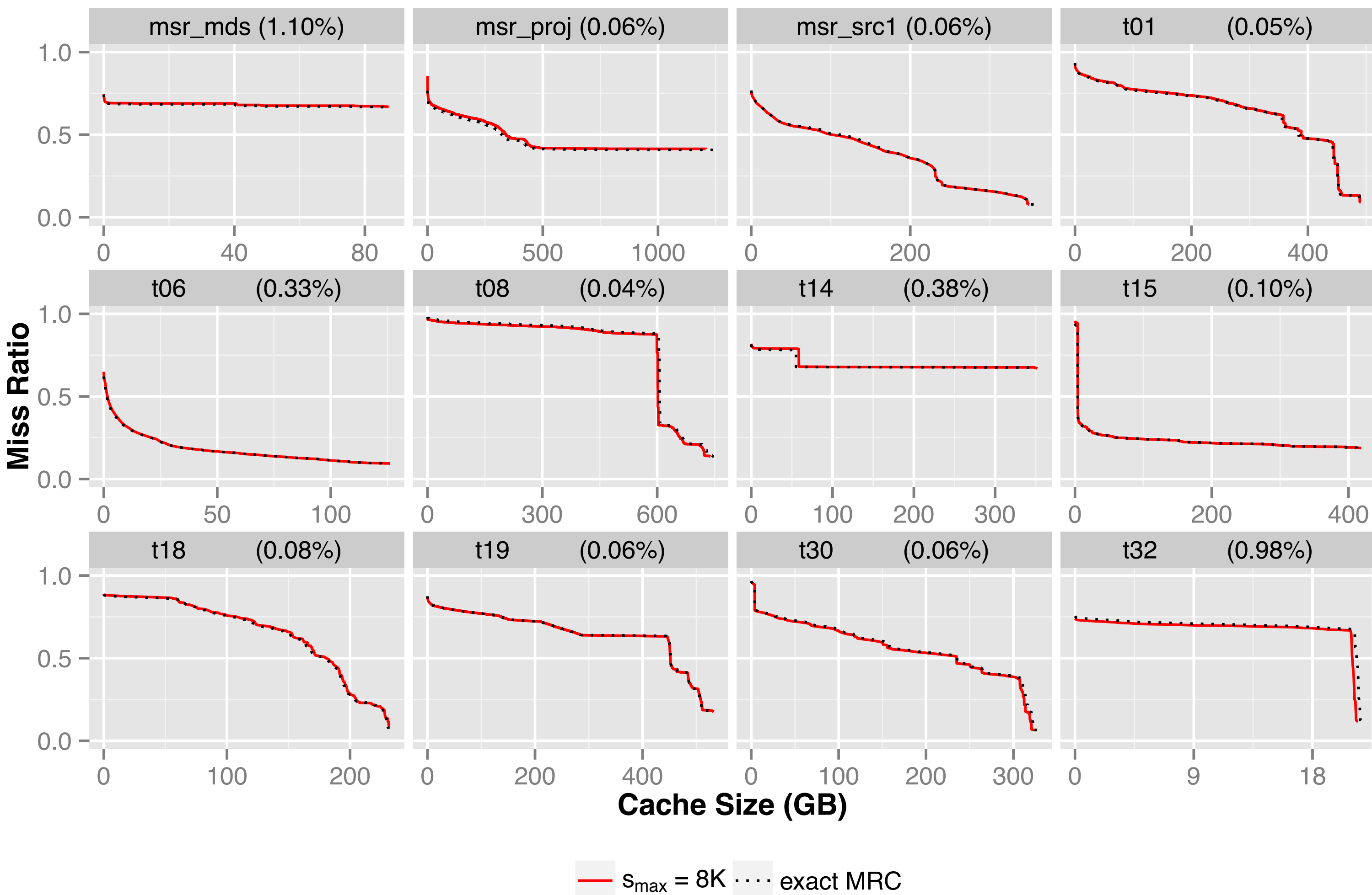
- Adjust sampling rate
  - Start with R = 0.1
  - Lower R as M increases
  - Shape depends on trace
- Rescale histogram counts
  - Discount evicted samples
  - Correct relative weighting
  - Scale by R_{new} / R_{old}

Experimental Evaluation

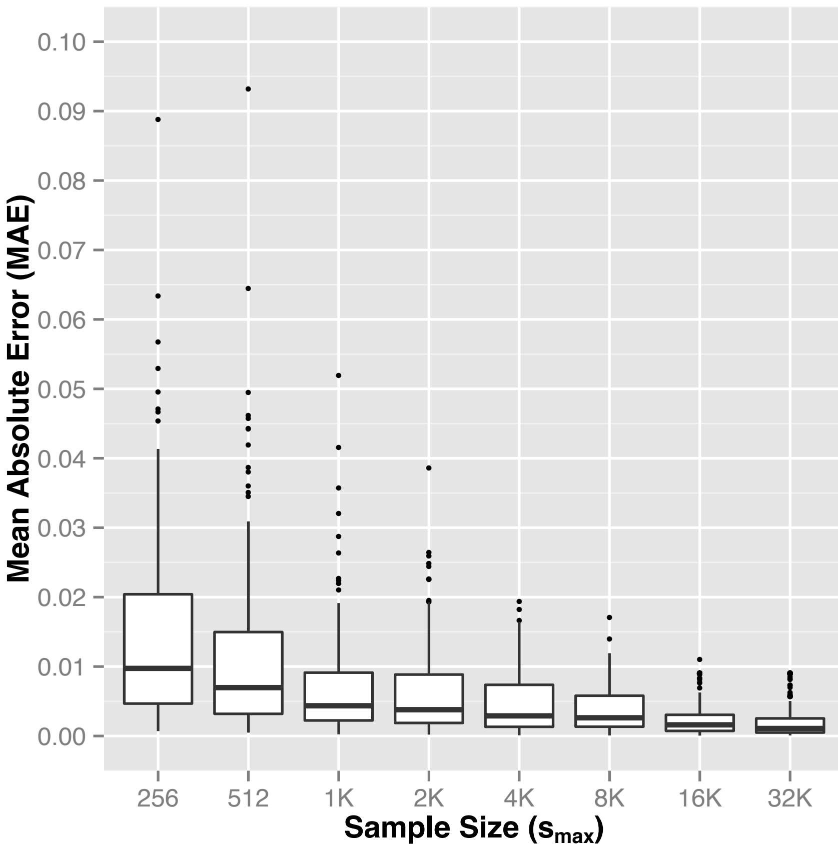


- Data collection
  - SaaS caching analytics
  - Remotely stream VMware vscsiStats
- 124 trace files
  - 106 week-long traces CloudPhysics customers
  - 12 MSR and 6 FIU traces SNIA IOTTA
- LRU, 16 KB block size

Exact MRCs vs. SHARDS

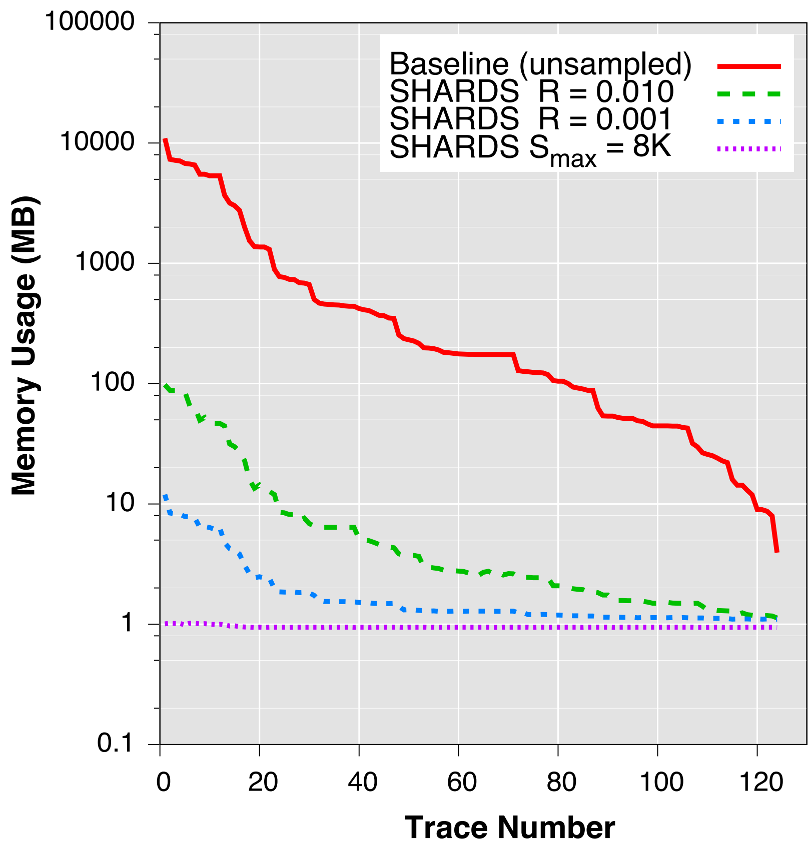


Error Analysis



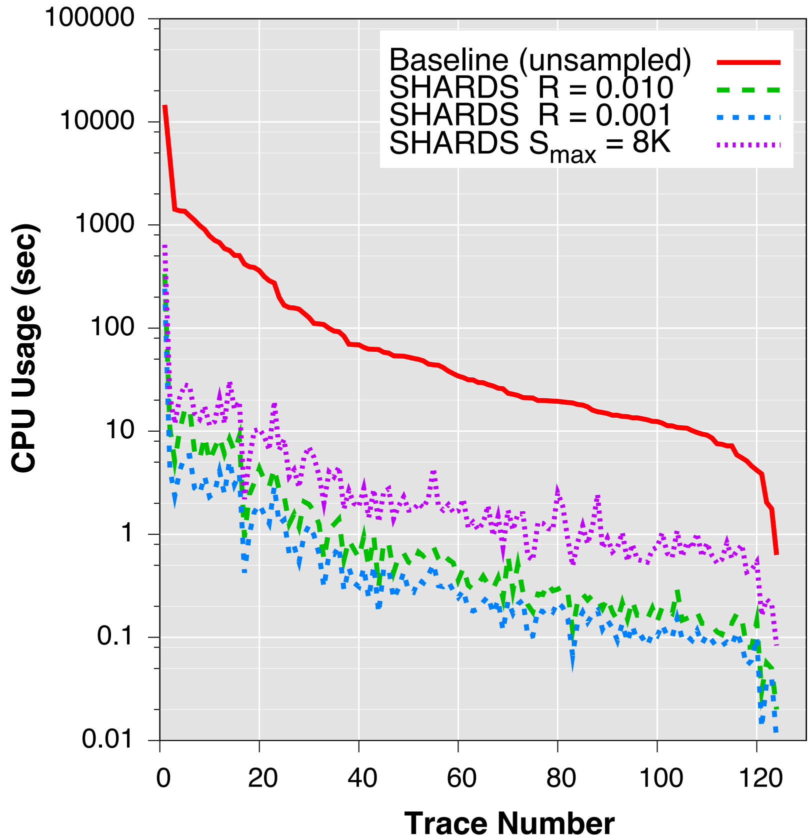
- Mean Absolute Error (MAE)
  - | exact – approx |
  - Average over all cache sizes
- Full set of 124 traces
- Error $\propto 1 / \sqrt{s_{max}}$
- MAE for s_{max} = 8K
  - 0.0027 median
  - 0.0171 worst-case

Memory Footprint



- Full set of 124 traces
- Sequential PARDA
- Basic SHARDS
  - Modified PARDA
  - Memory ≈ R × baseline for larger traces
- Fixed-size SHARDS
  - New space-efficient code
  - Constant 1 MB footprint

Processing Time



- Full set of 124 traces
- Sequential PARDA
- Basic SHARDS
  - Modified PARDA
  - R=0.001 speedup 41–1029×
- Fixed-size SHARDS
  - New space-efficient code
  - Overhead for evictions
  - S_{max}= 8K speedup 6–204×

Counter Stacks Comparison

Algorithm	Memory (MB)	Throughput (Mrefs/sec)	Error (MAE)
Counter Stacks	80.0	2.3	0.0025
SHARDS S_{max}=32K	2.0	16.9	0.0026
SHARDS S_{max}=8K	1.3	17.6	0.0061

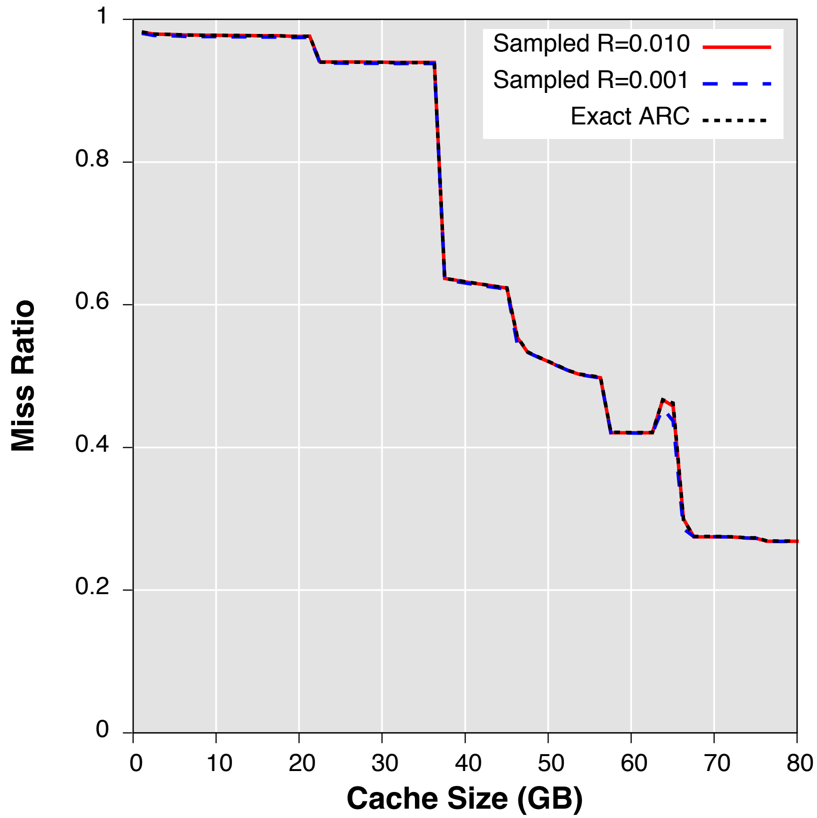
- Quantitative
  - Same merged MSR “master” trace
  - Counter Stacks roughly 7× slower, 40–62× bigger
- Qualitative
  - Counter Stacks checkpoints support splicing/merging
  - SHARDS maintains block ids, generalizes to non-LRU

Generalizing to Non-LRU Policies

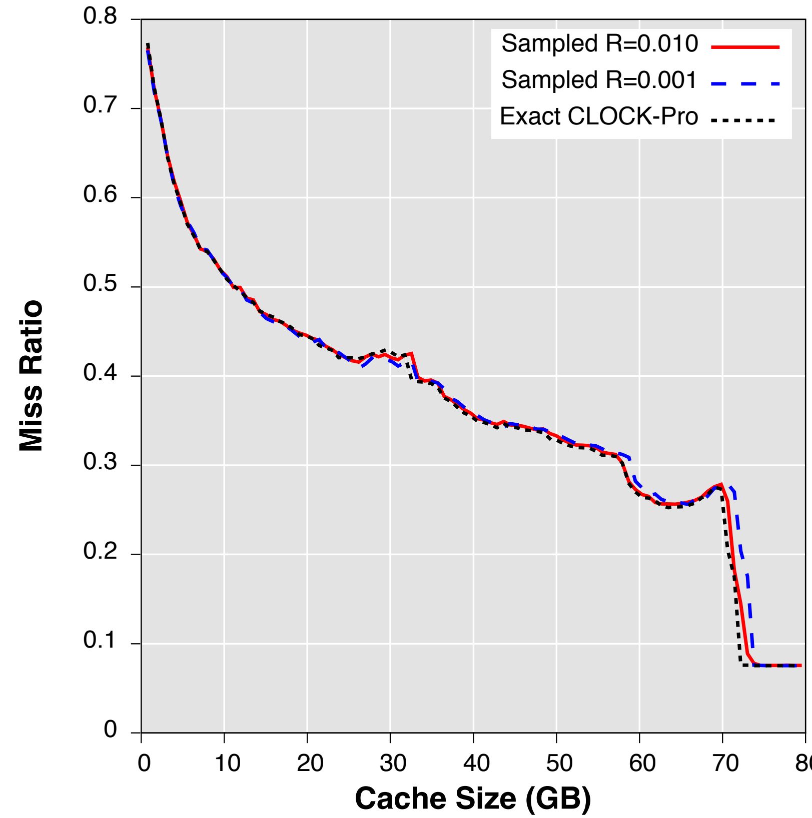
- Many sophisticated replacement policies
  - ARC, LIRS, CAR, CLOCK-Pro, …
  - Adaptive, frequency and recency
  - No known single-pass MRC methods!
- Solution: efficient scaled-down simulation
  - Filter using spatially hashed sampling
  - Scale down simulated cache size by sampling rate
  - Run full simulation at each cache size
- Surprisingly accurate results

Scaled-Down Simulation Examples

ARC — MSR-Web Trace



CLOCK-Pro — Trace *t04*



Conclusions

- New SHARDS algorithm
  - Approximate MRC in O(1) space, O(N) time
  - Excellent accuracy in 1 MB footprint
- Practical online MRCs
  - Even for memory-constrained drivers, firmware
  - So lightweight, can run multiple instances
- Scaled-down simulation of non-LRU policies

Questions?

- {carl,nohhyun,alex,irfan}@cloudphysics.com
- Visit our poster
- BoF 9-10pm tonight in Bayshore West
- Potential academic and industry collaboration
- Application areas include capacity planning, dynamic partitioning, tuning, policies, …
- We're also hiring!