An Efficient Simulation Algorithm for Cache of Random Replacement Policy

Shuchang Zhou

Institute of Computing Technology, Chinese Academy of Sciences

NPC'10 — Sep 13, 2010

Cache Mechanism: Cache Hit (1/3)

What's at a?

Cache

Address	**Content**
a	Hello
b	World
c	!

Cache Mechanism: Cache Hit (2/3)

What's at a?

Cache

Address	**Content**
a	Hello
b	World
c	!

Cache Mechanism: Cache Hit (3/3)

What's at a?

Cache

Address	**Content**
a	Hello
b	World
c	!

"Hello"

Cache Mechanism: Cache Miss (1/4)

What's at d?

Cache

Address	**Content**
a	Hello
b	World
c	!

Cache Mechanism: Cache Miss (2/4)

What's at d?

Cache

Address	**Content**
a	Hello
b	World
c	!

Backup mechanism

Cache Mechanism: Cache Miss (3/4)

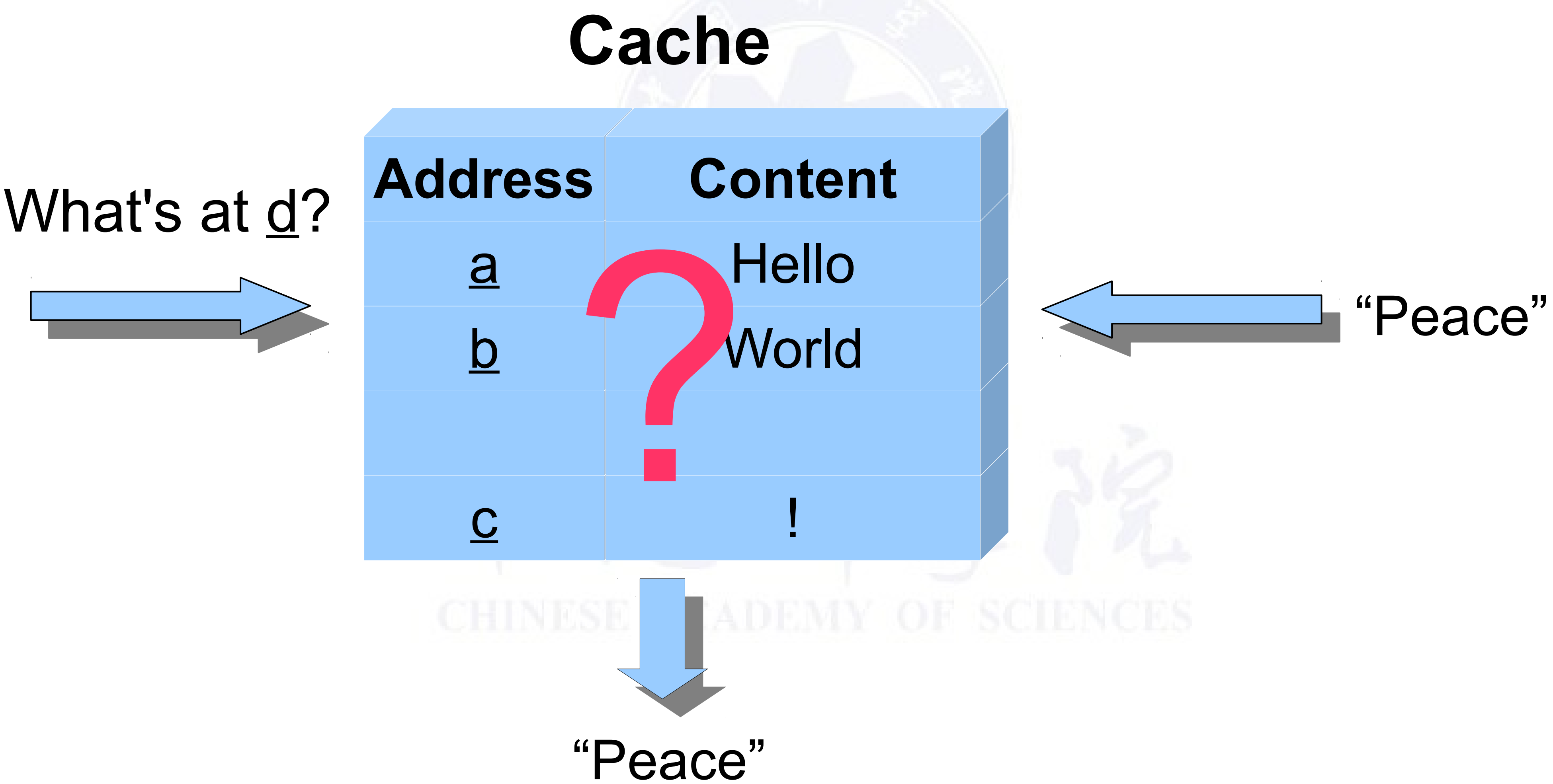
Cache

What's at d?

Address	**Content**
a	Hello
b	World
c	!

“Peace”

Cache Mechanism: Cache Miss (4/4)



Cache Mechanism: Replacement Policy

Option 1: Least Recently Used

- Evict the oldest data in cache
- Found in Intel/AMD processors

Option 2: Random Replacement

- Select randomly among candidates
- Found in ARM/Loongson processors

Cache of Random Replacement Policy (1/4)

Cache

Address	Content
a	Hello
b	World
c	!

Cache

Address	Content
d	Peace
b	World
c	!

Cache of Random Replacement Policy (2/4)

Cache

Address	Content
a	Hello
b	World
c	!

Cache

Address	Content
d	Peace
b	World
c	!

Cache

Address	Content
a	Hello
d	Peace
c	!

Cache of Random Replacement Policy (3/4)

Cache

Address	Content
a	Hello
b	World
c	!

Cache

Address	Content
d	Peace
b	World
c	!

Cache

Address	Content
a	Hello
d	Peace
c	!

Cache

Address	Content
a	Hello
b	World
d	Peace
c	!

Cache of Random Replacement Policy (4/4)

Cache

Address	Content
a	Hello
b	World
c	!

Cache

Address	Content
d	Peace
b	World
c	!

Cache

Address	Content
a	Hello
d	Peace
c	!

Cache

Address	Content
a	Hello
b	World
d	Peace
c	!

Cache

Address	Content
a	Hello
b	World
d	Peace

Random Replacement: Different Behavior (1/2)

Cache

What's at a?

Address	Content
d	Peace
b	World
c	!

Miss

Random Replacement: Different Behavior (2/2)

Cache

What's at a?

Address	Content
d	Peace
b	World
c	!

Miss

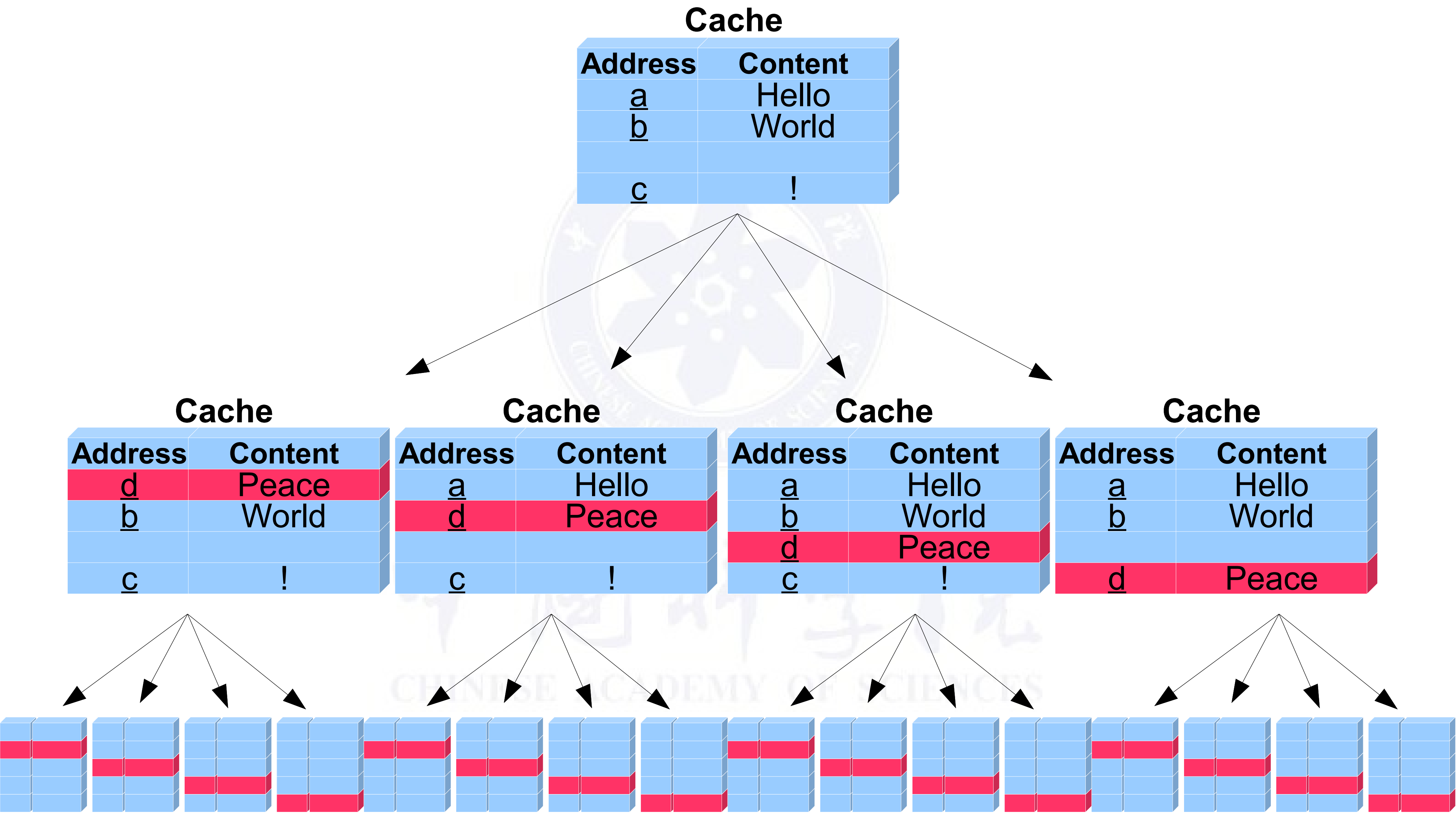
Cache

What's at a?

Address	Content
a	Hello
b	World
d	Peace
c	!

Hit

Random Replacement: State Explosion Problem (1/2)



Random Replacement: State Explosion Problem (2/2)

Solution 1

- Many rounds of Monte Carlo

Solution 2

- Reformulate the problem in stochastic setting

Background: Indicator Random Variable

Definition

- $X = 1$, if event happens
- $X = 0$, otherwise

Property

- The expectation equals the probability of the event
  - $E(X) = 1 * P(X=1) + 0 * P(X=0) = P(X=1)$

Stochastic Cache Simulation: Settings

Input

- Sequence of cache line index

  $a_0, a_1, a_2, \ldots a_n$

Cache Parameters

- Fully-associative
  - Extends to set-associative later
- Let M be the associativity
  - Hence there are M candidates for eviction

Stochastic Cache Simulation: Miss Event Indicators

Input

- Sequence of cache line index

  $a_0, a_1, a_2, \ldots a_n$

Miss Events

$X_0, X_1, X_2, \ldots X_n$

Stochastic Cache Simulation: Reuse Window

Definition

- From this access to last access to the same address

a,b,a,c,d,b

Stochastic Cache Simulation: Reuse Distance

Definition

- ∞ if the address never occurs before
- sum of X_i in the reuse window
- Dependent on associativity

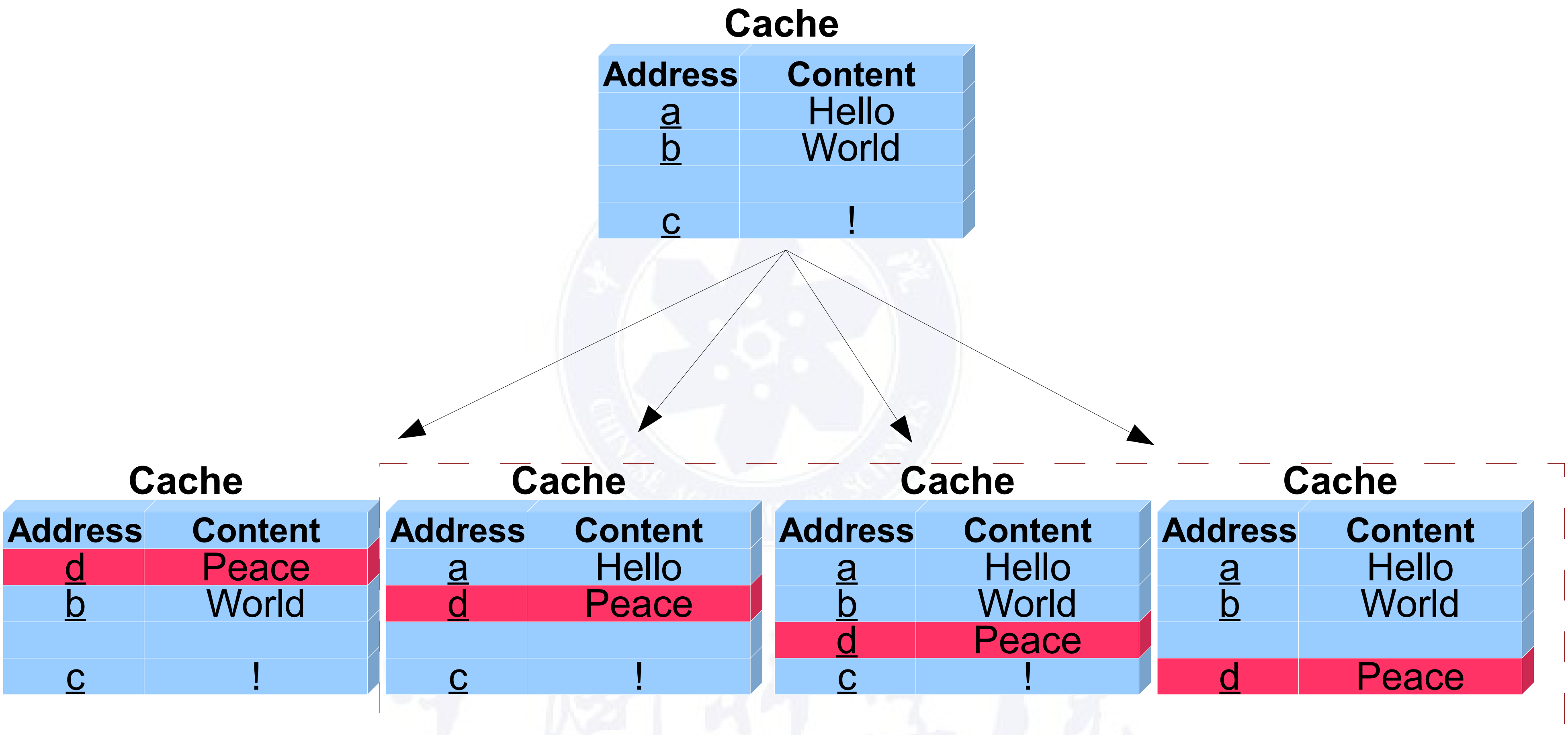
a,b,a,c,d,b

Stochastic Cache Simulation: Observations

Observations

- A cache line is definitely in cache after the access
- Every cache miss will have 1/M probability of evicting an old cache line present in cache

Random Replacement: Recall



a evicted

a remains

Stochastic Cache Simulation: Formula (1/2)

Observations

- A cache line is definitely in cache after the access
- Every cache miss will have $1/M$ probability of evicting an old cache line present in cache

Formula

- If there are Z misses in the reuse window, the probability of a cache line still being in cache is

$$(1 - 1/M)^Z$$

Stochastic Cache Simulation: Formula (2/2)

Formula

- $E(X_i|Z_i) = P(X_i=1) = 1 - (1 - 1/M)^{Z_i}$
- $E(X_i) = 1 - E((1 - 1/M)^{Z_i})$

Problem

- Impossible to solve directly, for the correlation between consecutive miss events

Stochastic Cache Simulation: Approximation

Formula

- $E(X_i|Z_i) = 1 - (1 - 1/M)^{Zi}$
- $E(X_i) = 1 - E((1 - 1/M)^{Zi})$

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- We can solve the recurrence relation to approximate the probability of each miss event.
- Other approximations of different precision exists.

Stochastic Cache Simulation: Example (1/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i						
EX_i						

Stochastic Cache Simulation: Example (2/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞					
EX_i						

Stochastic Cache Simulation: Example (3/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞					
EX_i	1					

Stochastic Cache Simulation: Example (4/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞	∞				
EX_i	1	1				

Stochastic Cache Simulation: Example (5/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞	∞	1			
EX_i	1	1				

Stochastic Cache Simulation: Example (6/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Z_i)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞	∞	1			
EX_i	1	1	0.25			

Assume M=4

Stochastic Cache Simulation: Example (7/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞	∞	1	∞	∞	2.25
EX_i	1	1	0.25	1	1	0.48

Assume M=4

Stochastic Cache Simulation: Example (8/8)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

	a	b	a	c	d	b
EZ_i	∞	∞	1	∞	∞	2.25
EX_i	1	1	0.25	1	1	0.48

Assume M=4

#miss ≈ 4.73

Stochastic Cache Simulation: Implementation

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞

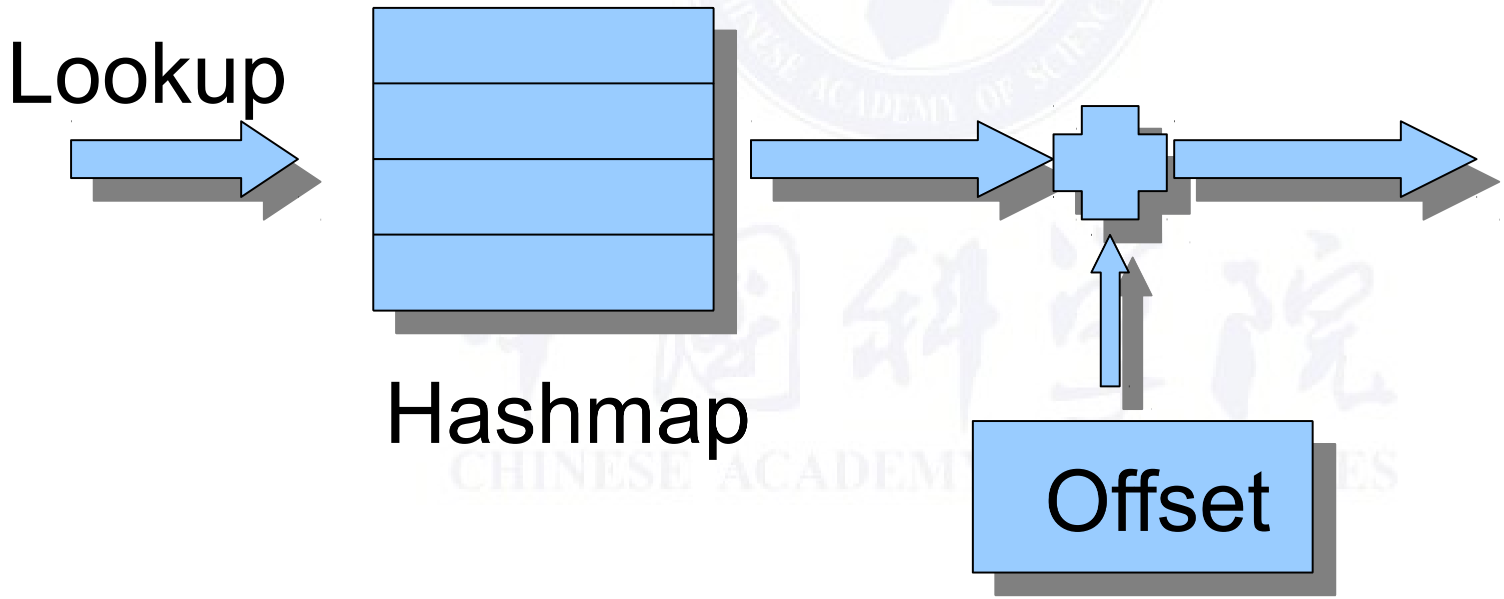
Maintenance

- Upon each cache miss, all EZ_i need be updated
- However, almost all incremented by the same value
  - except the one corresponding to a_i is set to 0

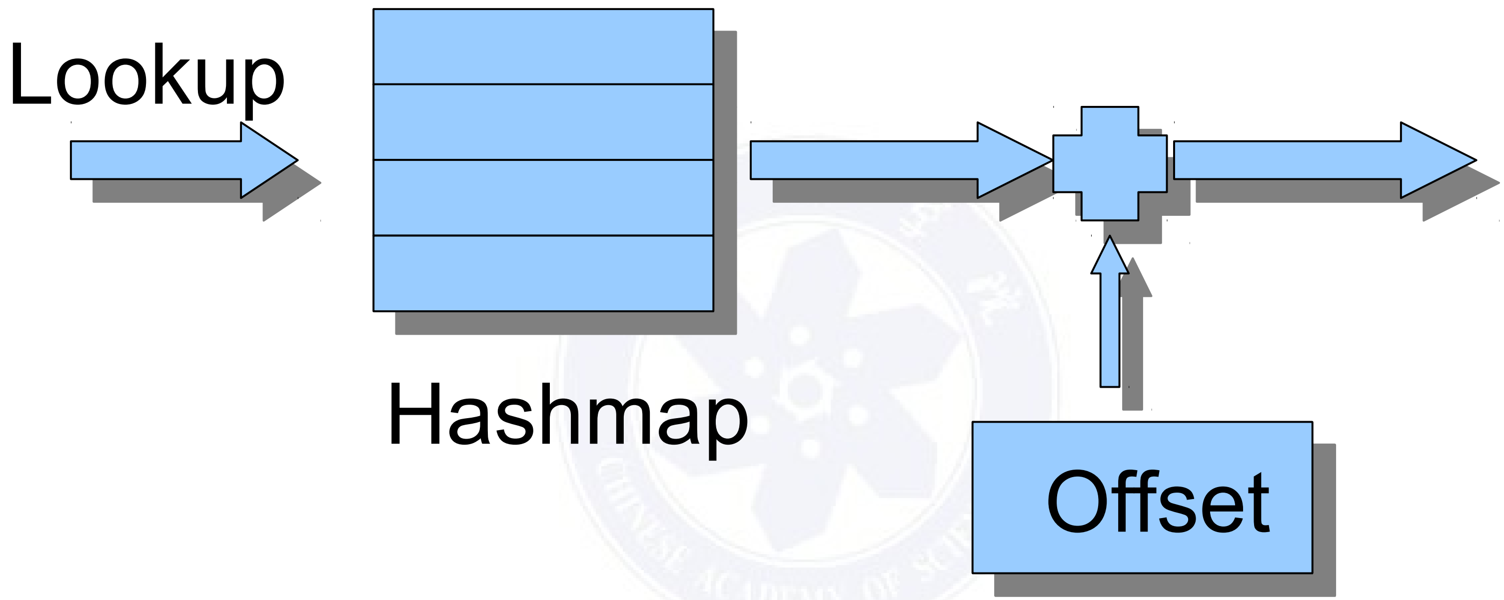
Hashmap with Offset: Structure (1/2)

Approximation

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Zi)}$
- $E(Z_i)$ = sum of $E(X_i)$ in the reuse window, or ∞



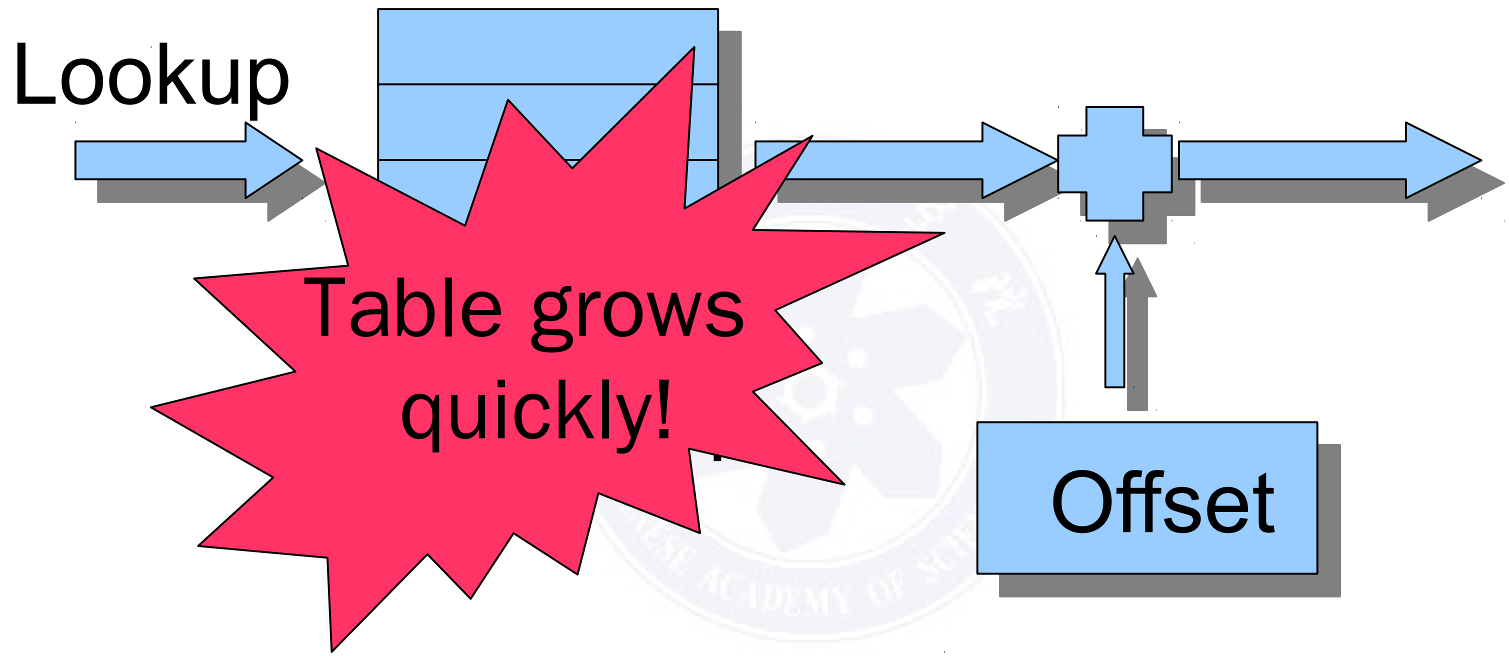
Hashmap with Offset: Structure (2/2)



Hashmap with Offset

- Stores EZ_i
- Absence from the hashmap indicates ∞
- $O(1)$ Lookup: Hashmap lookup + addition with offset
- $O(1)$ Maintenance: increment the offset and map a_i to 0

Hashmap with Offset: Space Problem



Hashmap with Offset

- Stores EZ_i
- Absence from the hashmap indicates ∞
- O(1) Lookup: Hashmap lookup + addition with offset
- O(1) Maintenance: increment the offset and map a_i to 0

Stochastic Cache Simulation: More Approximations

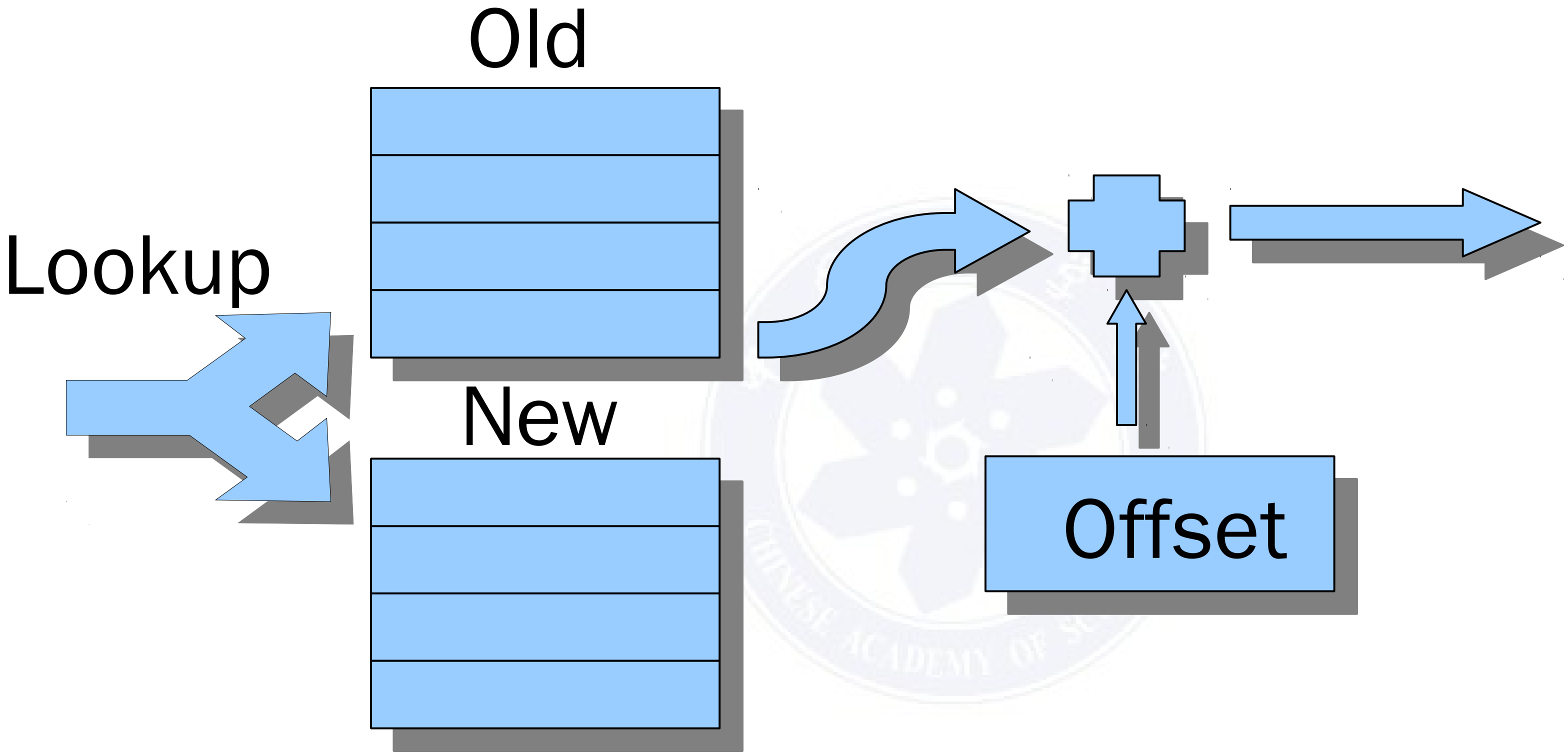
Approximation I

- $E(X_i) \approx 1 - (1 - 1/M)^{E(Z_i)}$

Approximation II

- Tolerate a little absolute error of the hit probability $1 - E(X_i)$
- Then can set large EZ_i to ∞
  - The same as removing them from the table

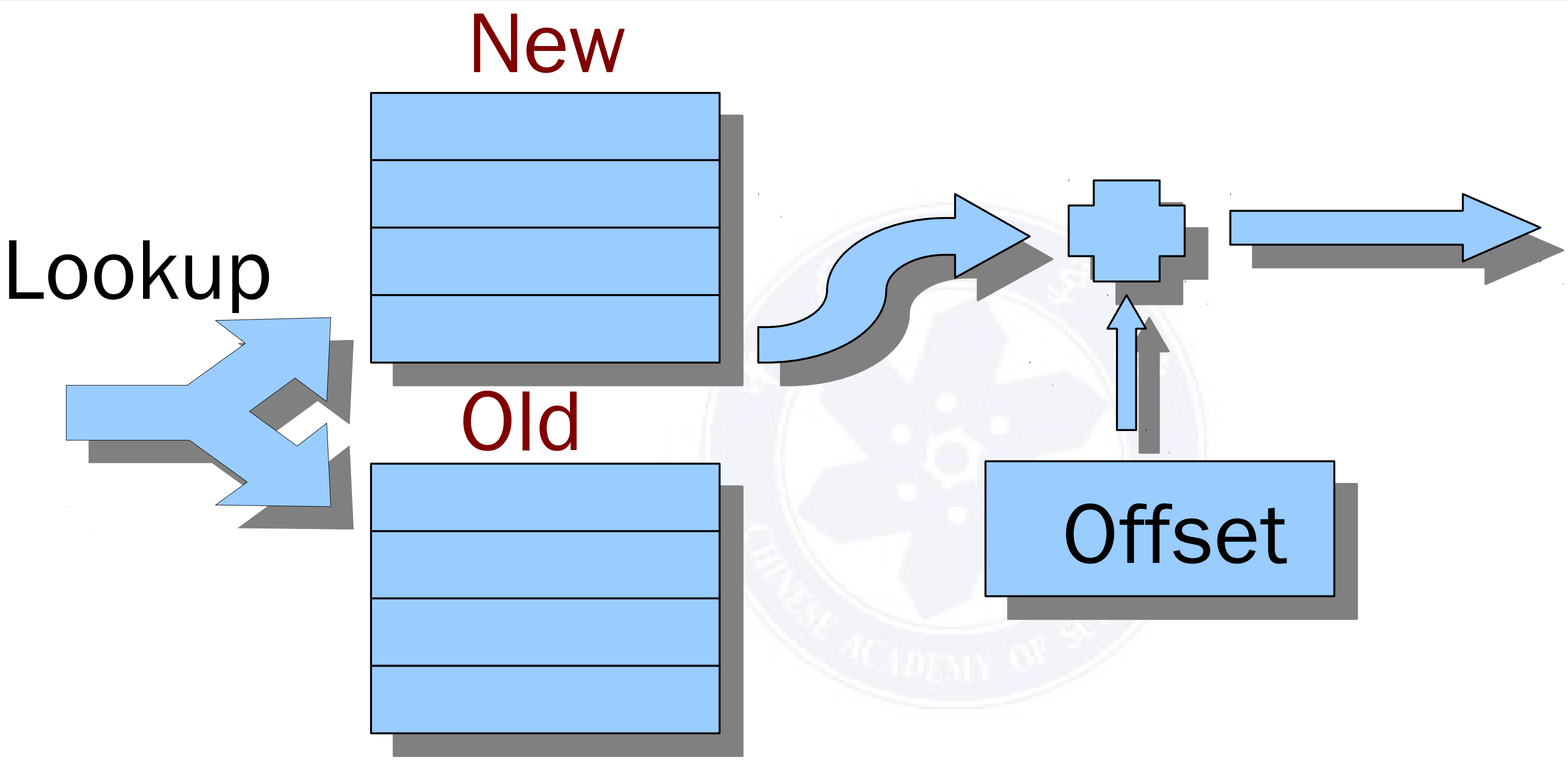
Hashmap with Offset: Sliding Windows (1/2)



Hashmap with Offset

- Stores EZ_i
- Absence from the hashmap indicates ∞
- Linear time lookup and maintenance
- Use sliding windows to control space

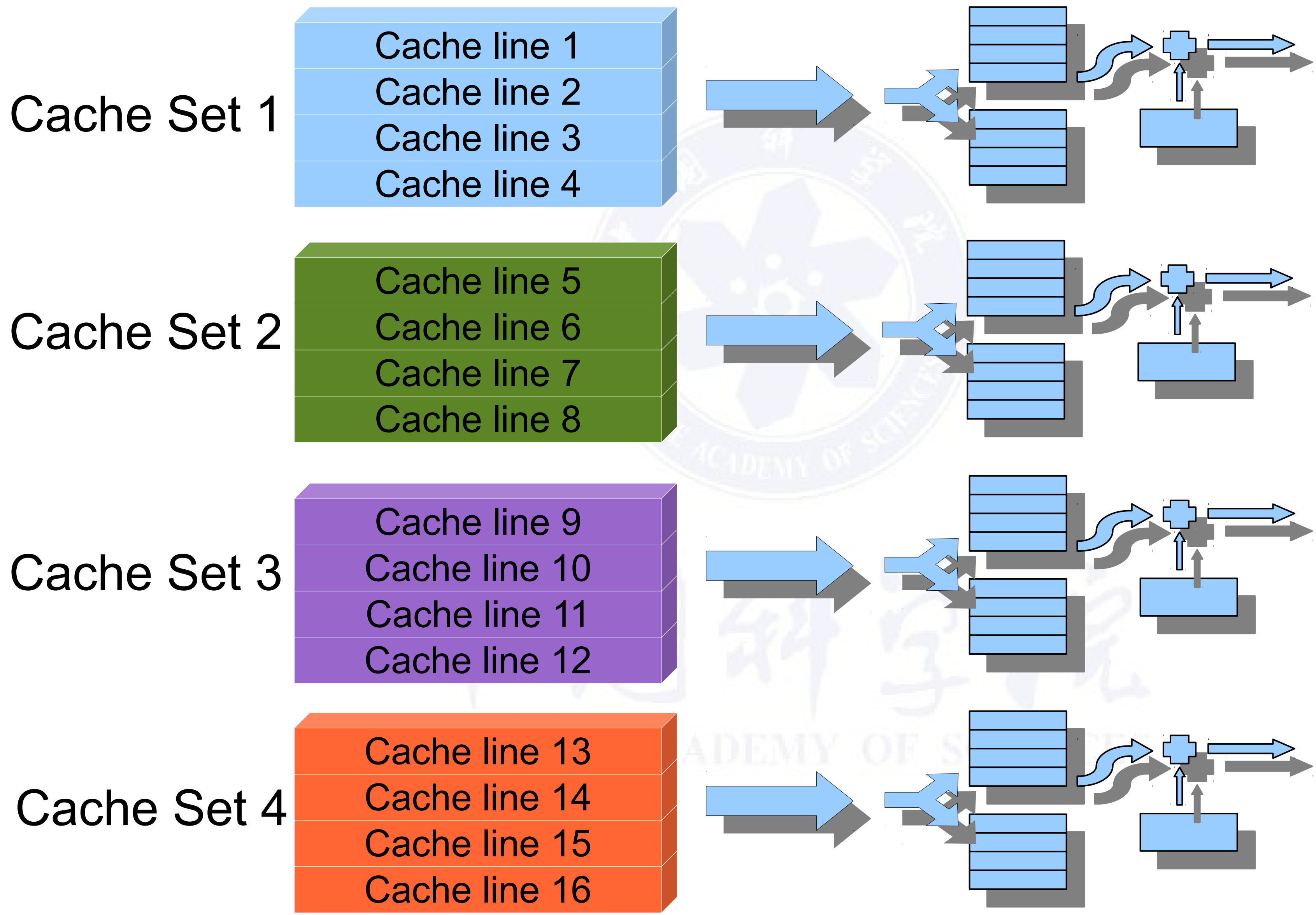
Hashmap with Offset: Sliding Windows (2/2)



Hashmap with Offset

- Stores EZ_i
- Absence from the hashmap indicates ∞
- Linear time lookup and maintenance
- Use sliding windows to control space

Hashmap with Offset: Set-associative cache



Hashmap with Offset: Summary

Hashmap with Offset

- Linear time lookup and maintenance
- Space: $\log \varepsilon / \log (1-1/M) \approx M \log(1/\varepsilon)$
- Precision: smaller with larger M
- Extends to set-associative cache

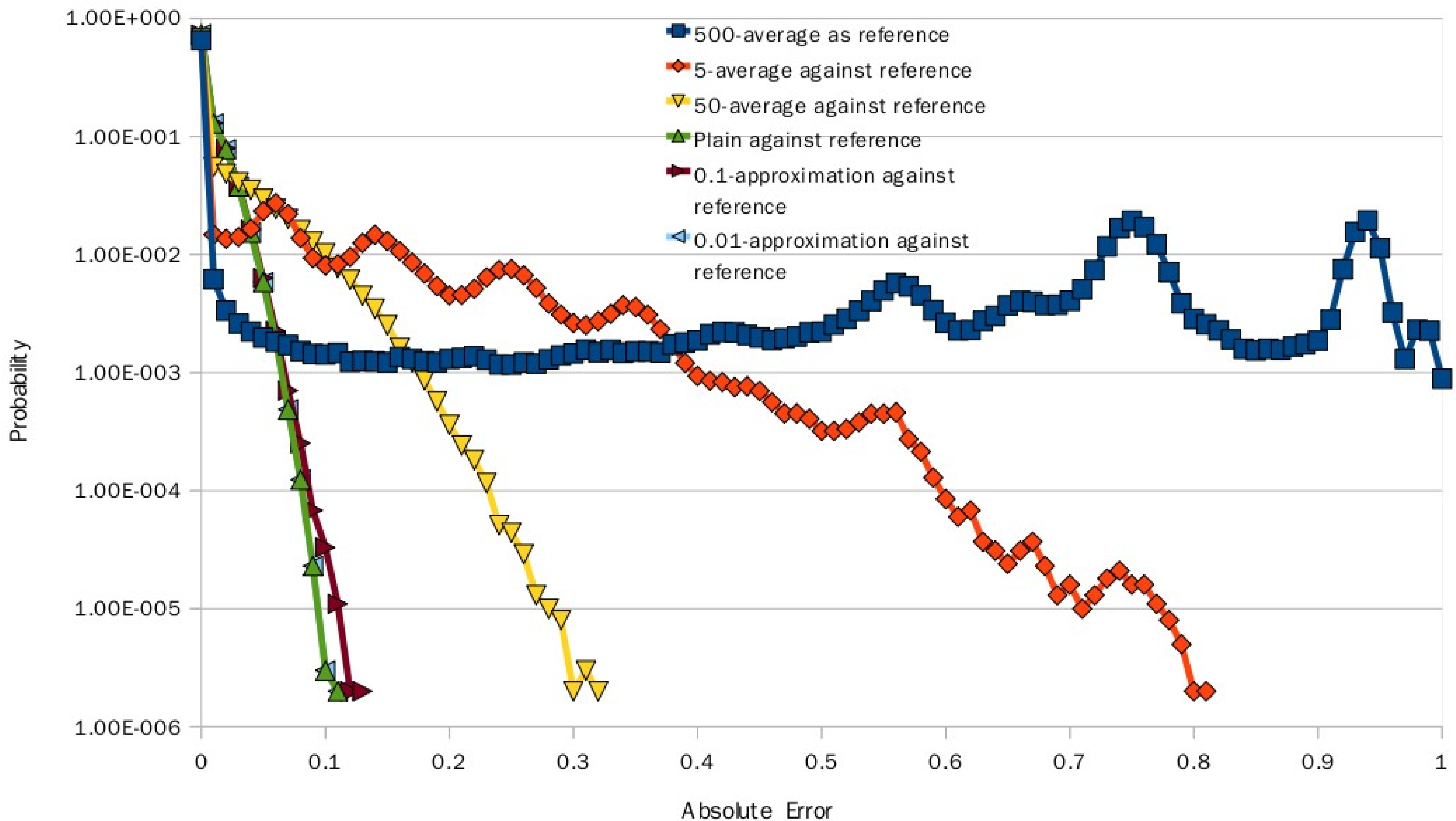
Empirical Evaluation: Setup

Setup

- Evaluated against multiple rounds of Monte Carlo simulation
  - Average of 500-rounds as reference
  - Compares the distribution of absolute errors
- Uses realistic traces collected by HMTT (Bao+08) for CPU2000/LINPACK

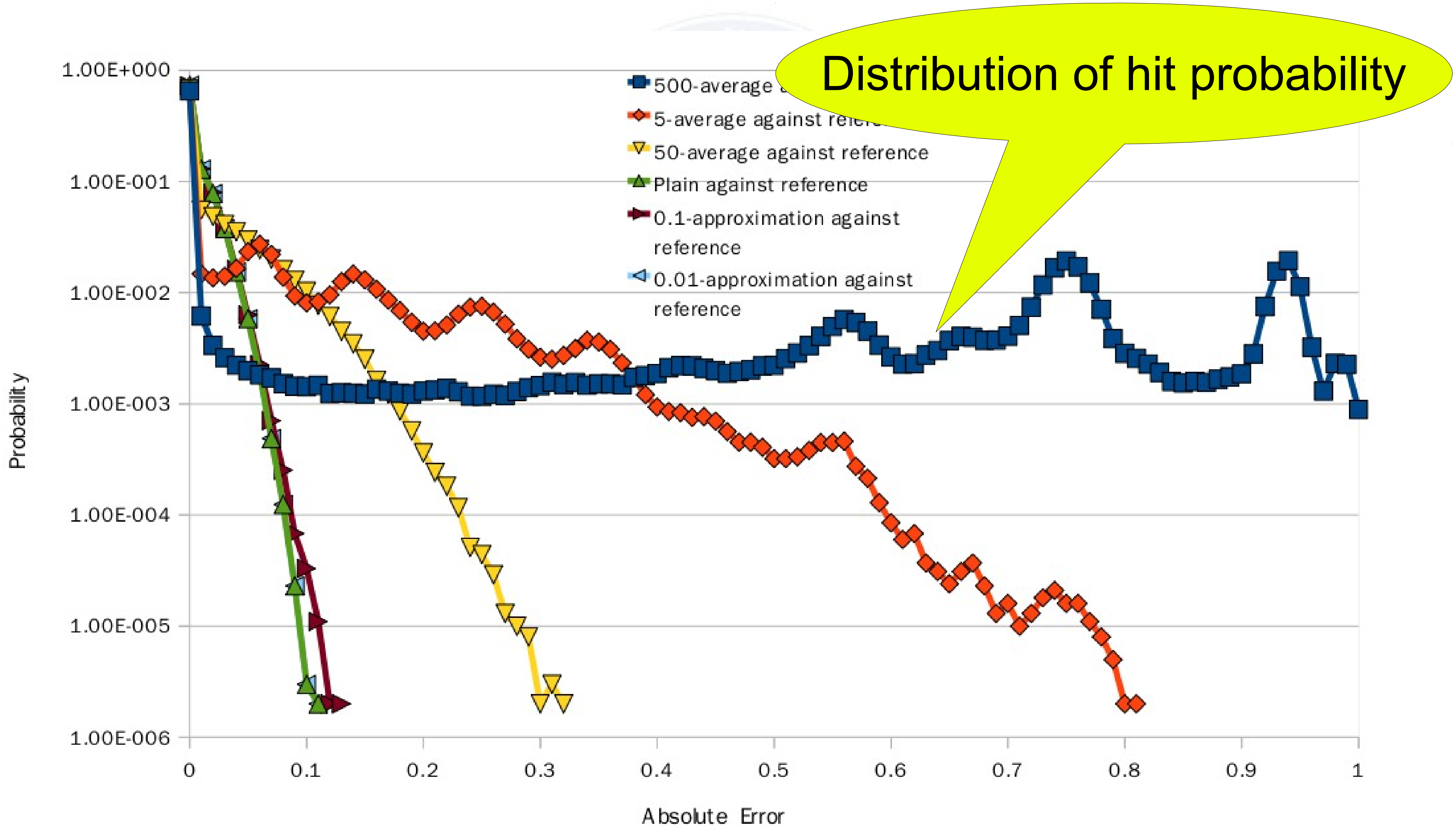
Empirical Evaluation: Precision (1/8)

M=4, CPU2000/SWIM



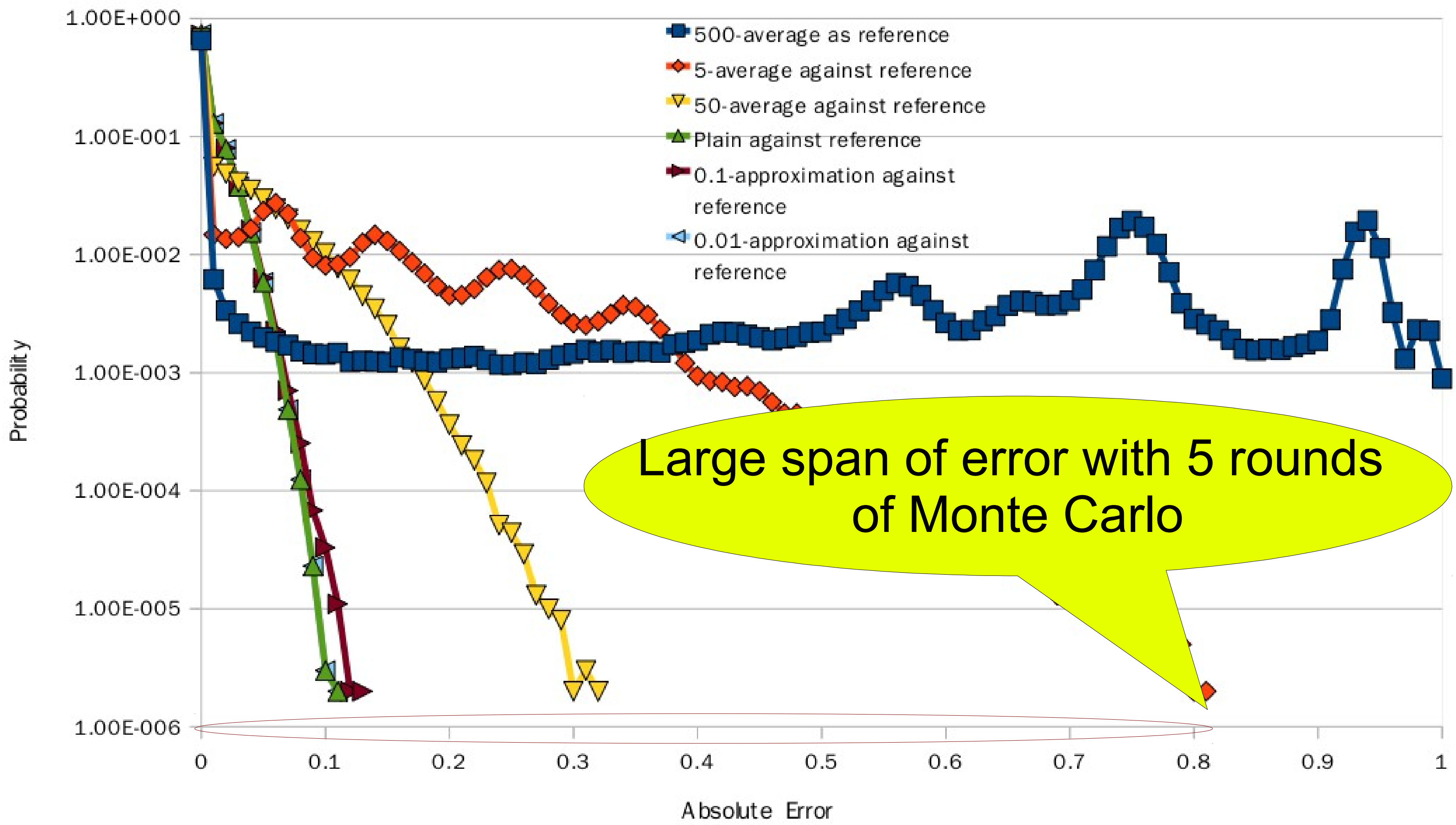
Empirical Evaluation: Precision (2/8)

M=4, CPU2000/SWIM



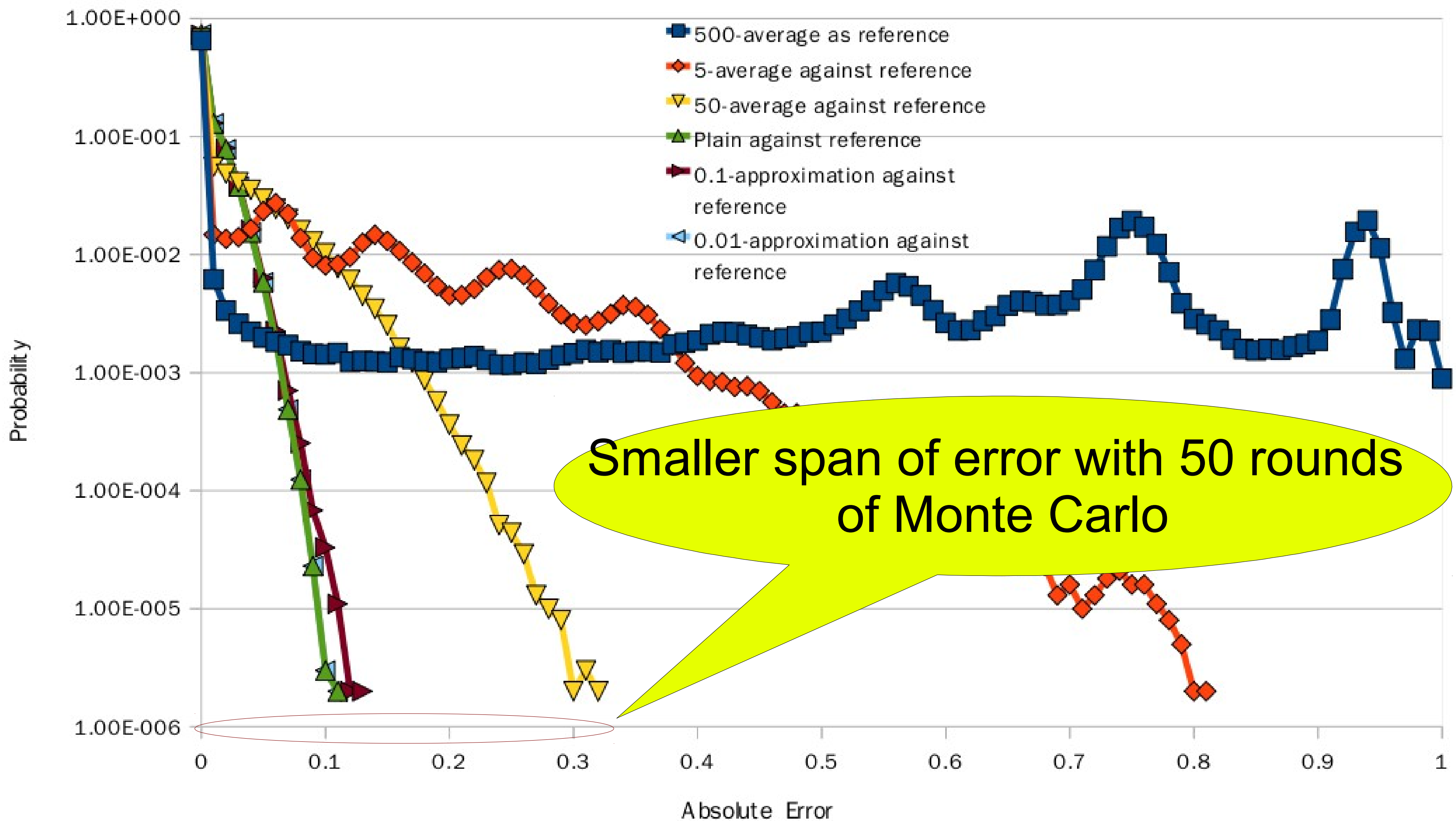
Empirical Evaluation: Precision (3/8)

M=4, CPU2000/SWIM



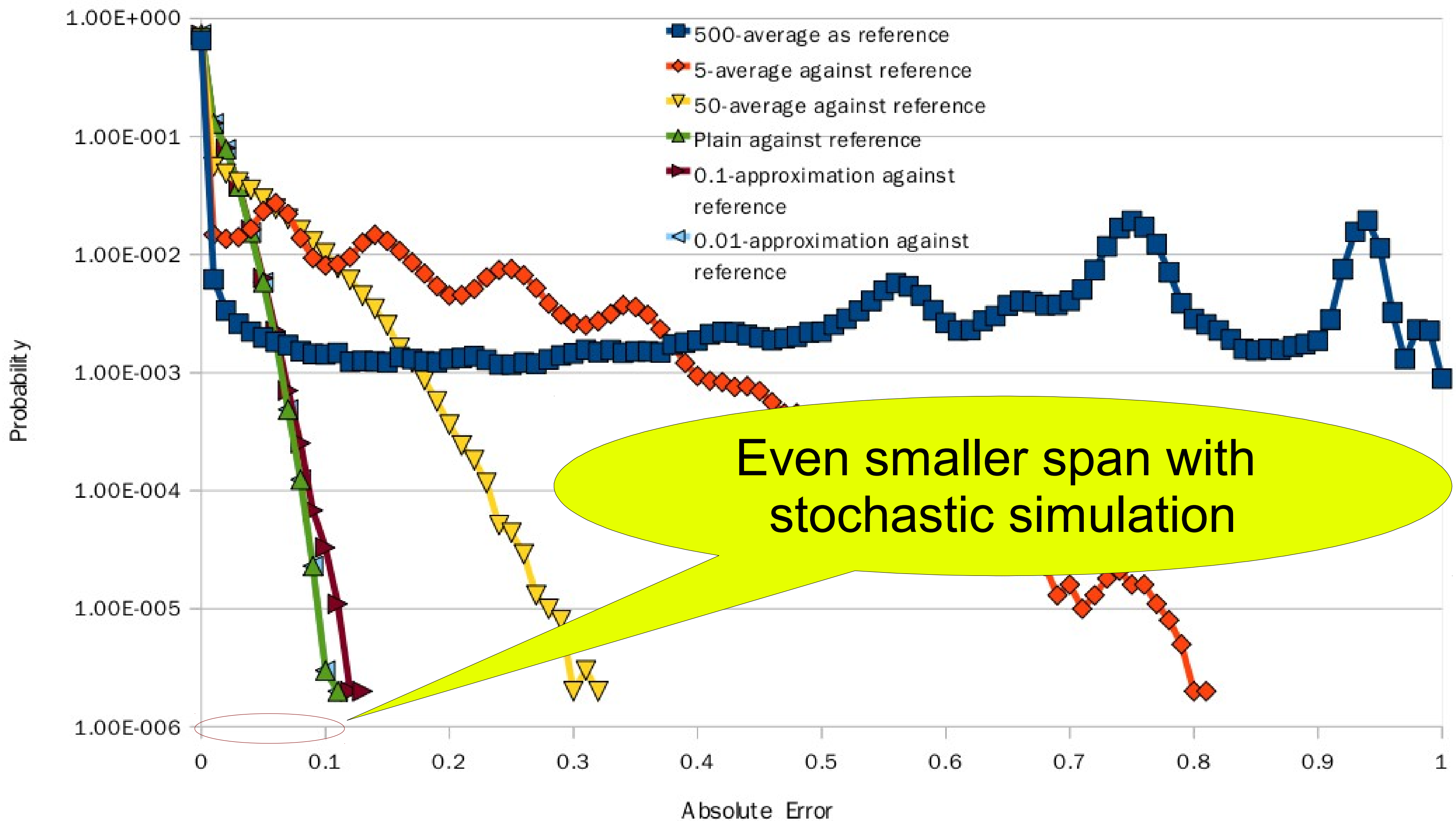
Empirical Evaluation: Precision (4/8)

M=4, CPU2000/SWIM



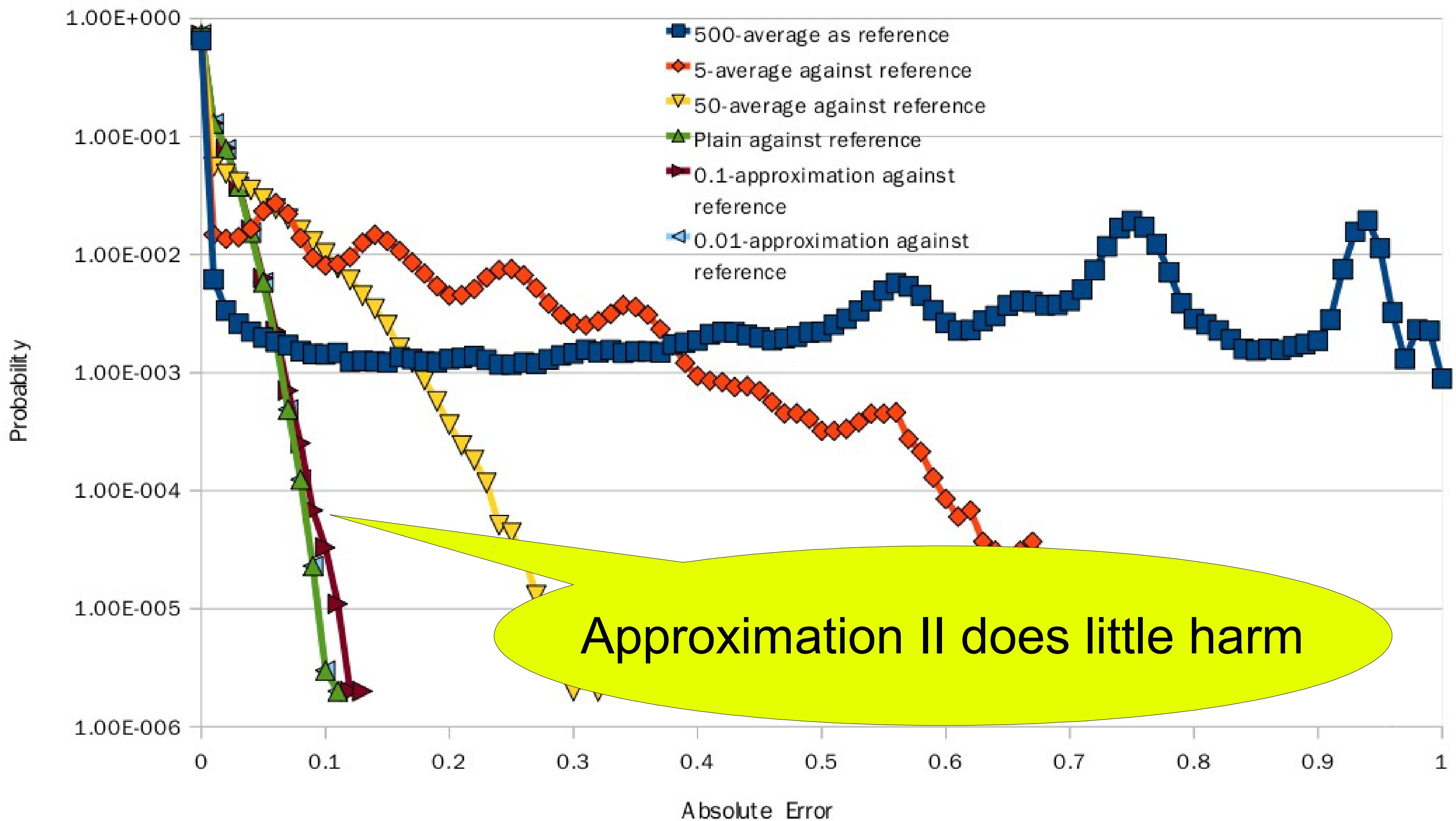
Empirical Evaluation: Precision (5/8)

M=4, CPU2000/SWIM



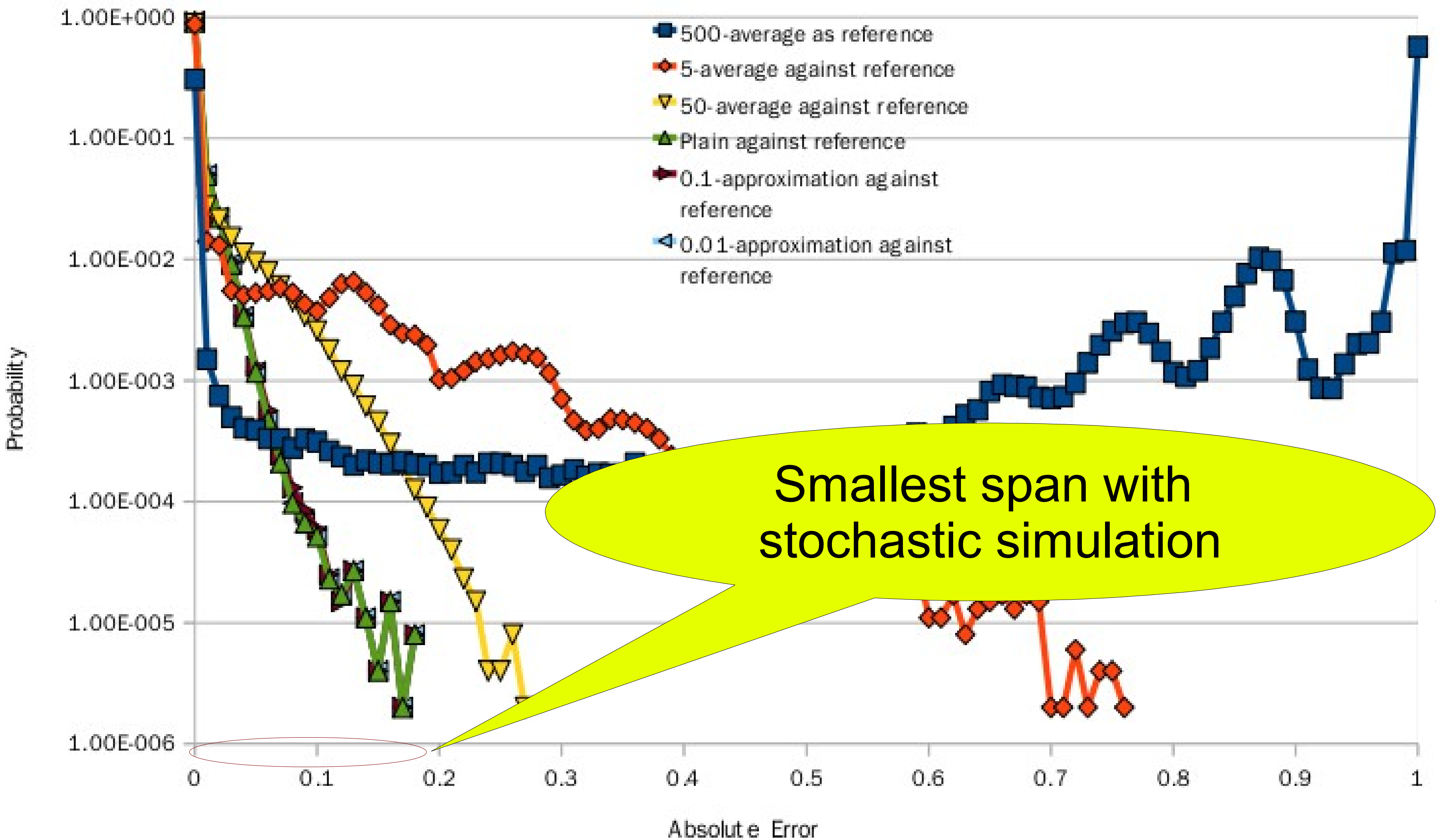
Empirical Evaluation: Precision (6/8)

M=4, CPU2000/SWIM



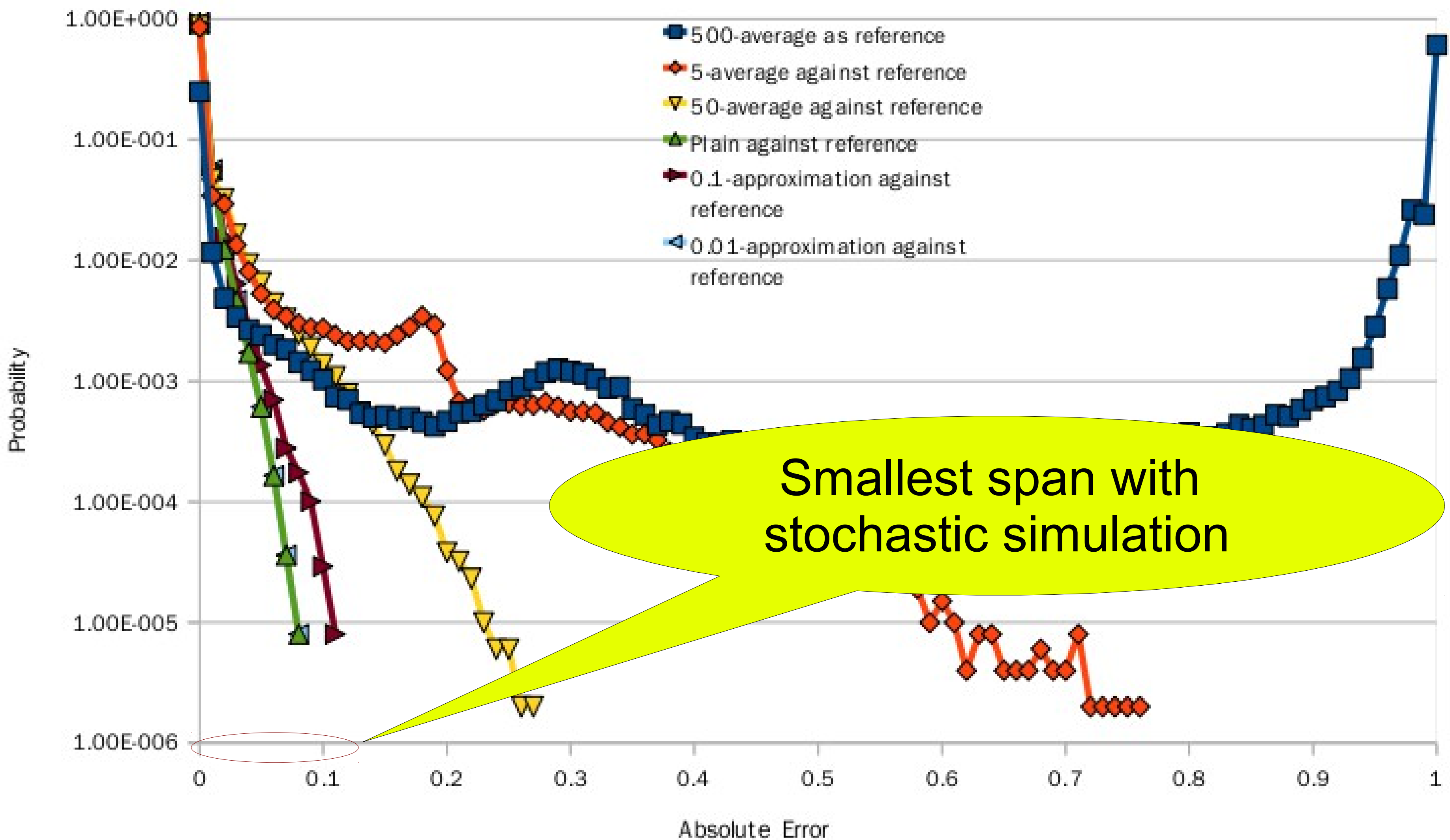
Empirical Evaluation: Precision (7/8)

M=8, LINPACK



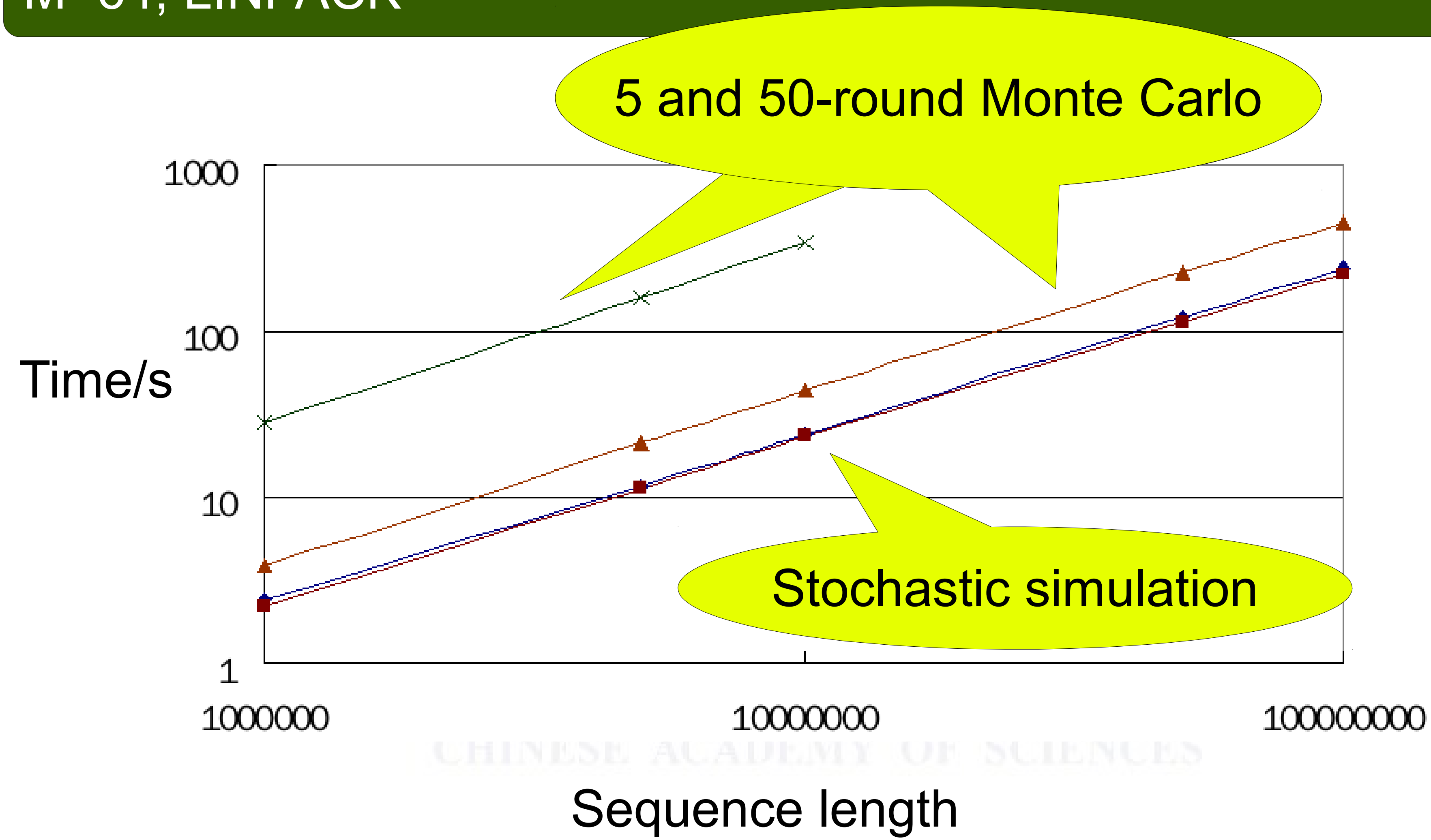
Empirical Evaluation: Precision (8/8)

M=64, LINPACK



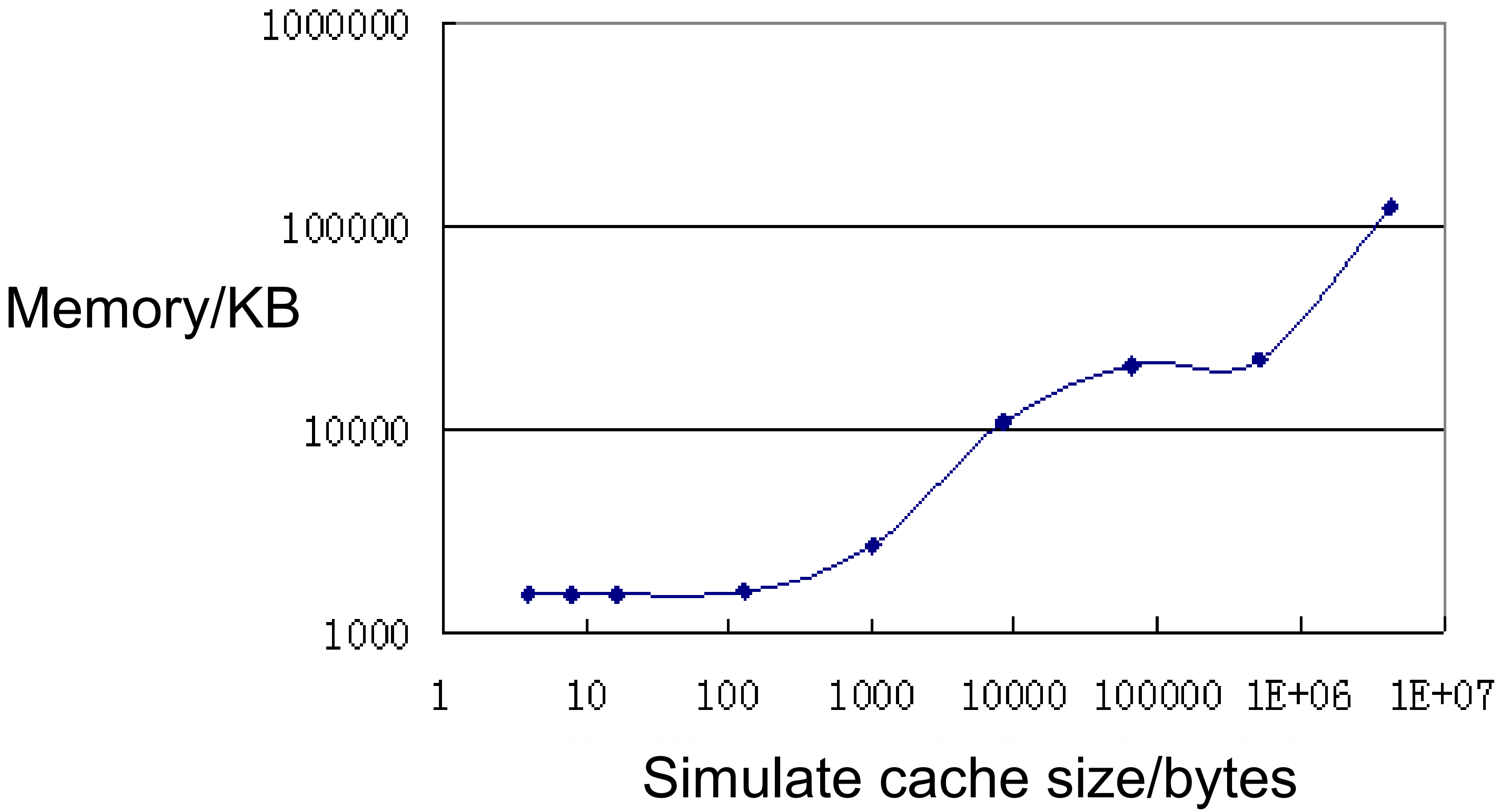
Empirical Evaluation: Time

M=64, LINPACK



Empirical Evaluation: Space

M=64, LINPACK, ε = 0.01



Conclusion

Contributions

- Stochastic simulation of Cache of Random Replacement policy
- Efficient Data structure with O(1) query and maintenance and almost linear space
- Extends to set-associative cache

Future work

- Application in Execution Time Analysis
- Extends to LRU cache

Q&A

Thank you!