Modeling Cache Sharing on Chip Multiprocessor Architectures

Pavlos Petoumenos[1], Georgios Keramidas[1], Håkan Zeffer[2], Stefanos Kaxiras[1], Erik Hagersten[2]

[1]University of Patras, Greece

[2]Uppsala University, Sweden

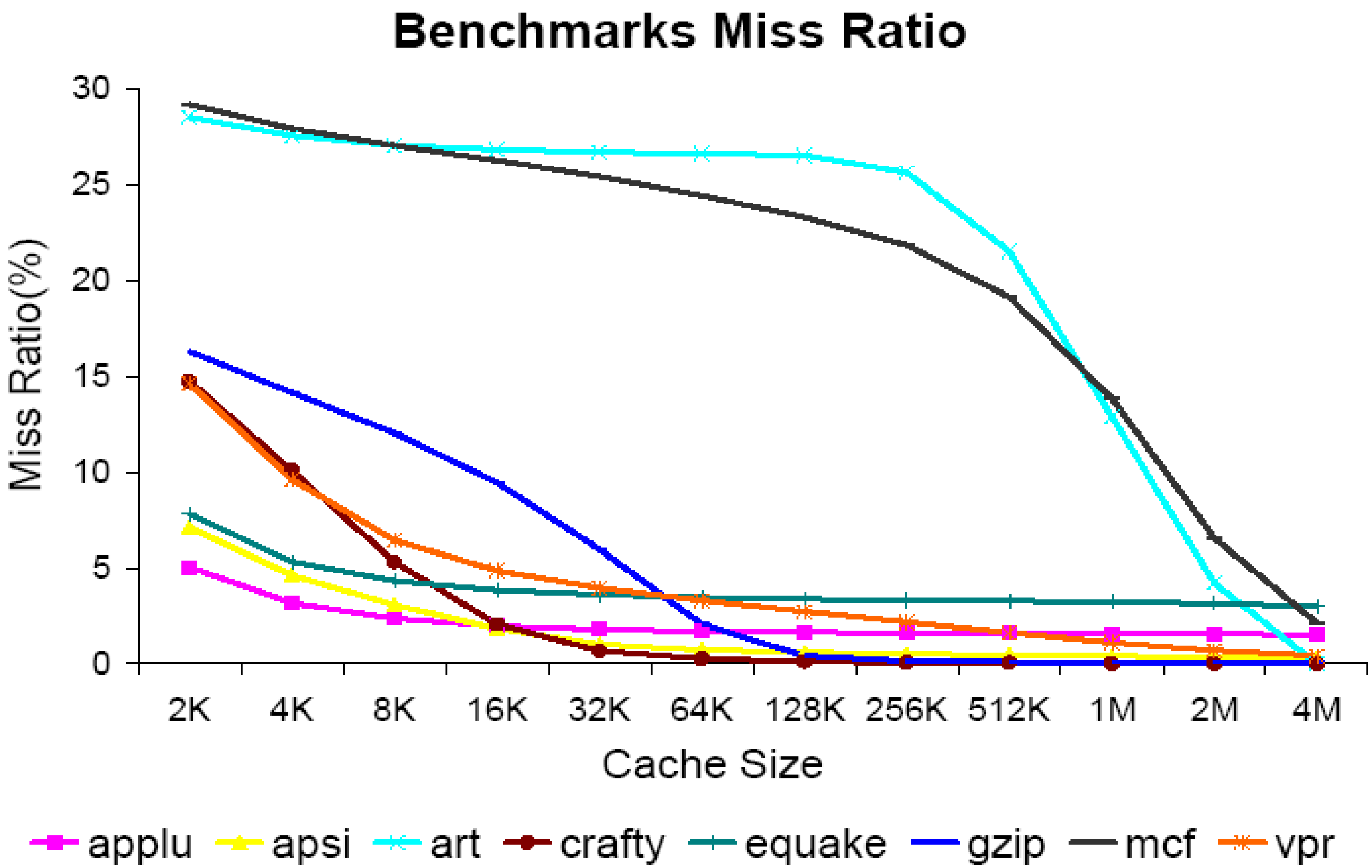
Introduction

- CMPs: the new dominant architecture
- Need to understand and manage CMP cache sharing
  - Performance
  - Fairness
  - QoS
  - DoS
- Contributions:
  - Accurate, online, statistical model describing sharing behavior
  - A control mechanism driven by the model

Our inspiration: Statcache

- Measure *reuse distances* of cache lines in accesses
- Simple model FA-RR cache:
  - Miss and hit probability for reuse distance *i as a function of miss rate*
- Solve *iteratively* for miss rate given a cache size
  - Miss rate for any cache
  - Only offline

Statcache output (motivation)



StatShare

- Like StatCache a statistical FA-RR cache model
- But models cache sharing of a single (Set Associative) cache
- In *CAT-time*
  - Time measured in Cache Allocation Ticks, i.e., replacements
  - Greatly simplifies equations ⇨ online model

StatShare model

h(.) app 1

h(.) app 2

•
•
•

h(.) app n

$$f(i) = 1 - \left(1 - \frac{1}{L}\right)^i$$

$$\overline{f}(i) = 1 - f(i)$$

$$l(i) = L \times f(i)$$

$$misses = \sum_{i=0}^{\infty} h(i)\, f(i)$$

$$hits = \sum_{i=0}^{\infty} h(i)\, \overline{f}(i)$$

$$S = \sum_{i=0}^{\infty} L \times h(i) \times f(i) = L \times misses$$

$$S_{hit} = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

$$S_{miss} = \sum_{i=0}^{\infty} \big((L + i) \times f(i) - i\big) \times h(i)$$

i: reuse distance

h(i): reuse distance histogram in CAT

f(i): miss probability function

l(i): life expectancy function

L: size of the cache in cachelines

S: Spacetime

StatShare model - Histograms

h(.) app 1

h(.) app 2

•
•
•

h(.) app n

$$f(i) = 1 - \left(1 - \frac{1}{L}\right)^{i}$$

$$\overline{f}(i) = 1 - f(i)$$

$$l(i) = L \times f(i)$$

$$misses = \sum_{i=0}^{\infty} h(i)\, f(i)$$

$$hits = \sum_{i=0}^{\infty} h(i)\, \overline{f}(i)$$

$$S = \sum_{i=0}^{\infty} L \times h(i) \times f(i) = L \times misses$$

$$S_{hit} = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

$$S_{miss} = \sum_{i=0}^{\infty} \big((L + i) \times f(i) - i\big) \times h(i)$$

i: reuse distance

h(i): reuse distance histogram in CAT

f(i): miss probability function

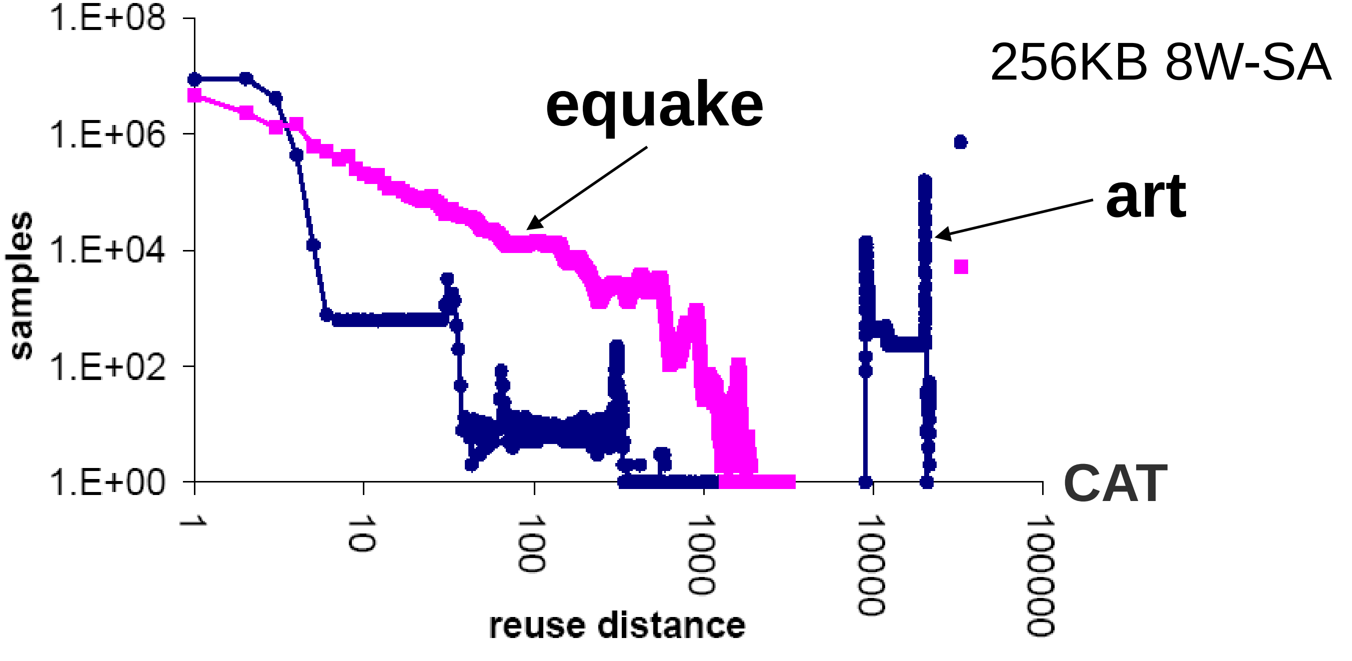
l(i): life expectancy function

L: size of the cache in cachelines

S: Spacetime

CAT reuse distance histograms

- *h(i)* : #accesses of a CAT reuse distance *i*
- Characterize the behavior of an app in a shared cache



StatShare model – f-functions

h(.) app 1

h(.) app 2

•
•
•

h(.) app n

$$f(i) = 1 - \left(1 - \frac{1}{L}\right)^{i}$$

$$\overline{f}(i) = 1 - f(i)$$

$$l(i) = L \times f(i)$$

$$misses = \sum_{i=0}^{\infty} h(i)\, f(i)$$

$$hits = \sum_{i=0}^{\infty} h(i)\, \overline{f}(i)$$

$$S = \sum_{i=0}^{\infty} L \times h(i) \times f(i) = L \times misses$$

$$S_{hit} = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

$$S_{miss} = \sum_{i=0}^{\infty} \big((L + i) \times f(i) - i\big) \times h(i)$$

i: reuse distance

h(i): reuse distance histogram in CAT

f(i): miss probability function

l(i): life expectancy function

L: size of the cache in cachelines

S: Spacetime

f-functions

- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache

f-functions

- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache

Probability of :

Being selected for replacement: 1/L

f-functions

- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache

Probability of :

Being selected for replacement: 1/L

Not being selected for replacement: 1-1/L

f-functions

- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache

Probability of :

Being selected for replacement: $1/L$

Not being selected for replacement: $1-1/L$

Remaining in the cache after *i* CAT: $(1- 1/L)^i$ = ***f-bar***

f-functions

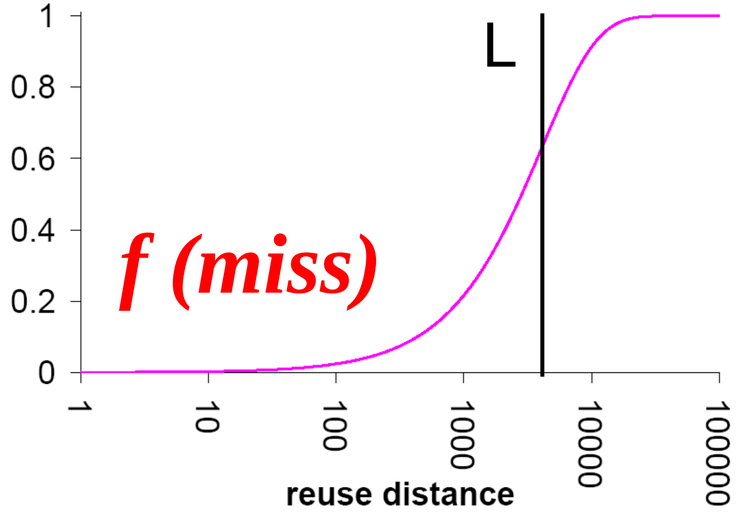
- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache

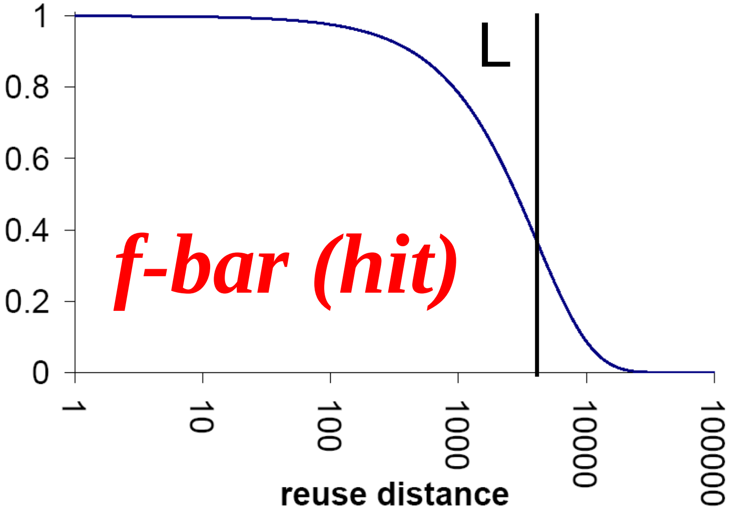
Probability of :

Being selected for replacement:	$1/L$
Not being selected for replacement:	$1-1/L$
Remaining in the cache after *i* CAT:	$(1- 1/L)^i$ = ***f-bar***
Having been replaced after *i* CAT:	$1-(1- 1/L)^i$ = ***f***

f-functions

- Probability functions of a miss or a hit for a histogram sample
  - with reuse distance *i*
  - in a *L*-line FA-RR cache





StatShare model – hits & misses

h(.) app 1

h(.) app 2

•
•
•

h(.) app n

$$f(i) = 1 - \left(1 - \frac{1}{L}\right)^{i}$$

$$\overline{f}(i) = 1 - f(i)$$

$$l(i) = L \times f(i)$$

$$misses = \sum_{i=0}^{\infty} h(i)\, f(i)$$

$$hits = \sum_{i=0}^{\infty} h(i)\, \overline{f}(i)$$

$$S = \sum_{i=0}^{\infty} L \times h(i) \times f(i) = L \times misses$$

$$S_{hit} = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

$$S_{miss} = \sum_{i=0}^{\infty} \big((L+i) \times f(i) - i\big) \times h(i)$$

i: reuse distance

h(i): reuse distance histogram in CAT

f(i): miss probability function

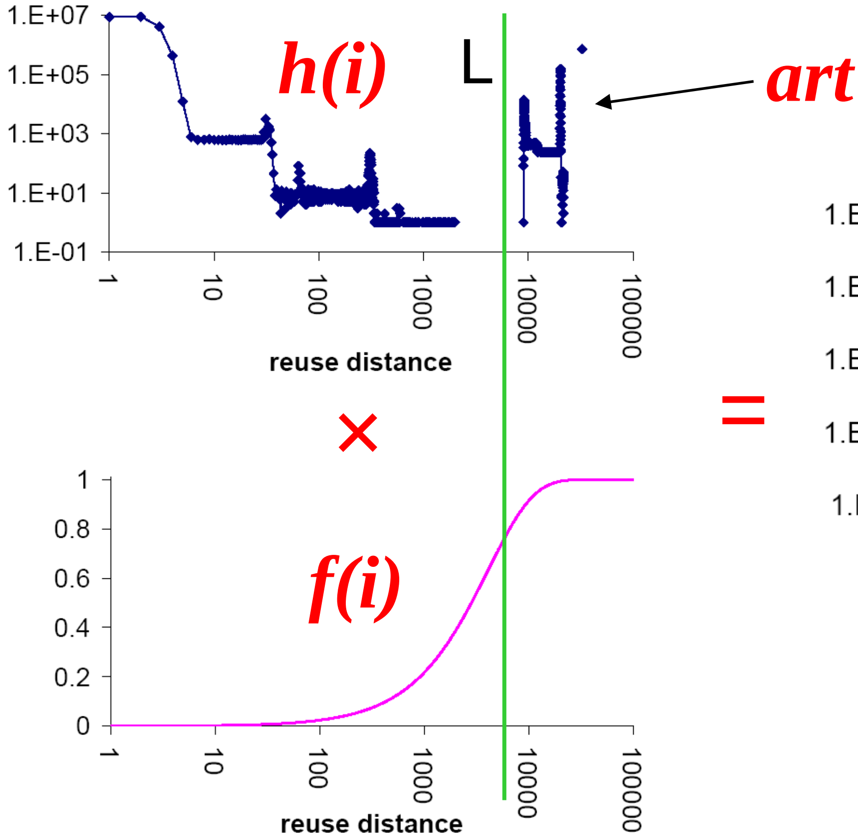
l(i): life expectancy function

L: size of the cache in cachelines

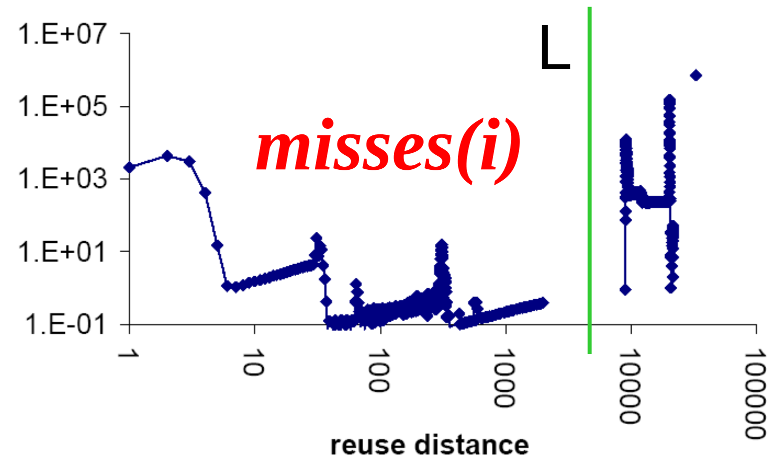
S: Spacetime

misses

- *h* multiplied by *f* gives us misses



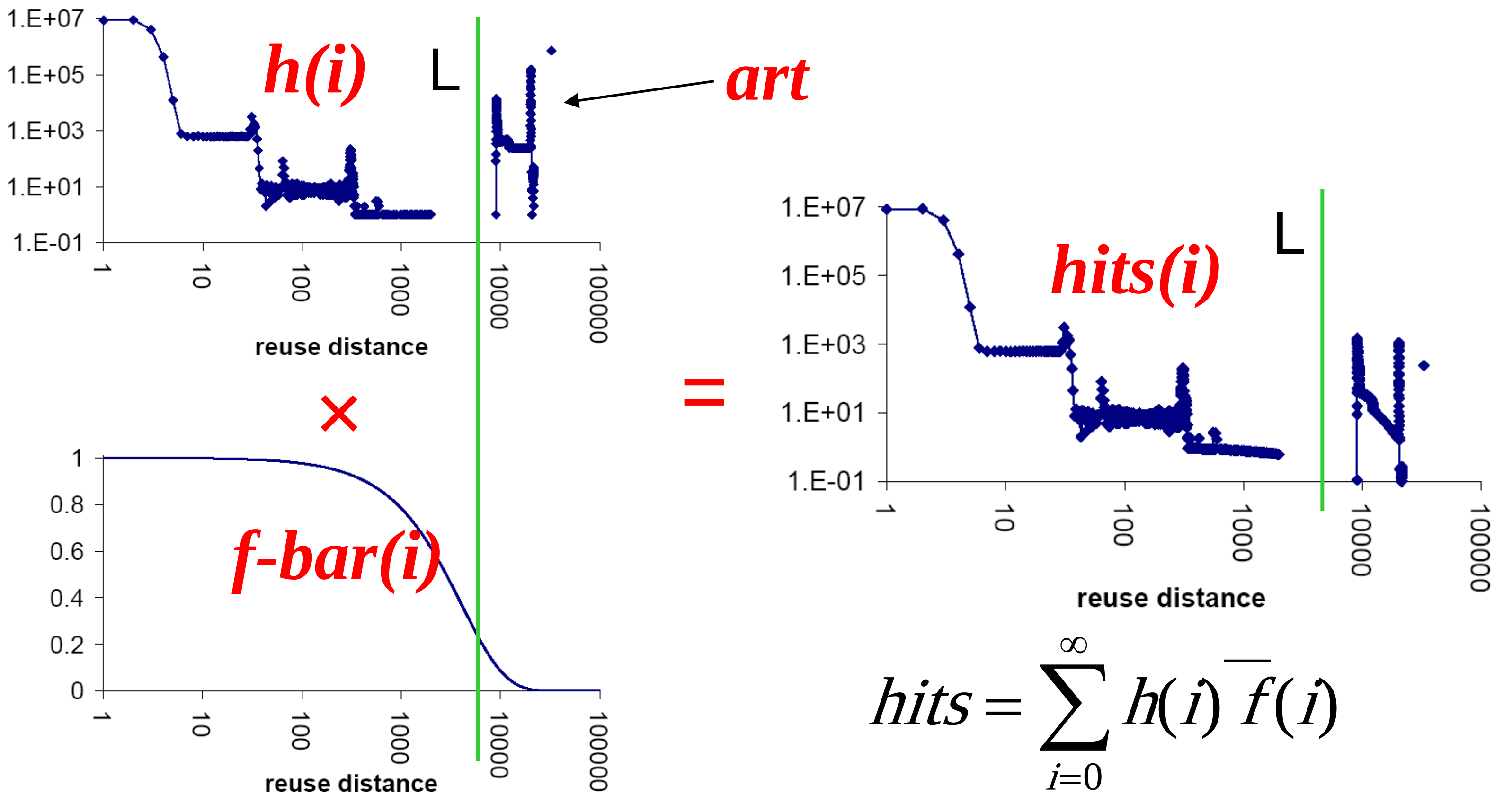
=



$$misses = \sum_{i=0}^{\infty} h(i)\ f(i)$$

hits

- *h* multiplied by *f-bar* gives us hits



$$hits = \sum_{i=0}^{\infty} h(i)\,\overline{f}(i)$$

StatShare model – Space time

h(.) app 1

h(.) app 2

...

h(.) app n

$$f(i) = 1 - \left(1 - \frac{1}{L}\right)^{i}$$

$$\overline{f}(i) = 1 - f(i)$$

$$l(i) = L \times f(i)$$

$$misses = \sum_{i=0}^{\infty} h(i)\, f(i)$$

$$hits = \sum_{i=0}^{\infty} h(i)\, \overline{f}(i)$$

$$S = \sum_{i=0}^{\infty} L \times h(i) \times f(i) = L \times misses$$

$$S_{hit} = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

$$S_{miss} = \sum_{i=0}^{\infty} \big((L + i) \times f(i) - i\big) \times h(i)$$

i: reuse distance

h(i): reuse distance histogram in CAT

f(i): miss probability function

l(i): life expectancy function

L: size of the cache in cachelines

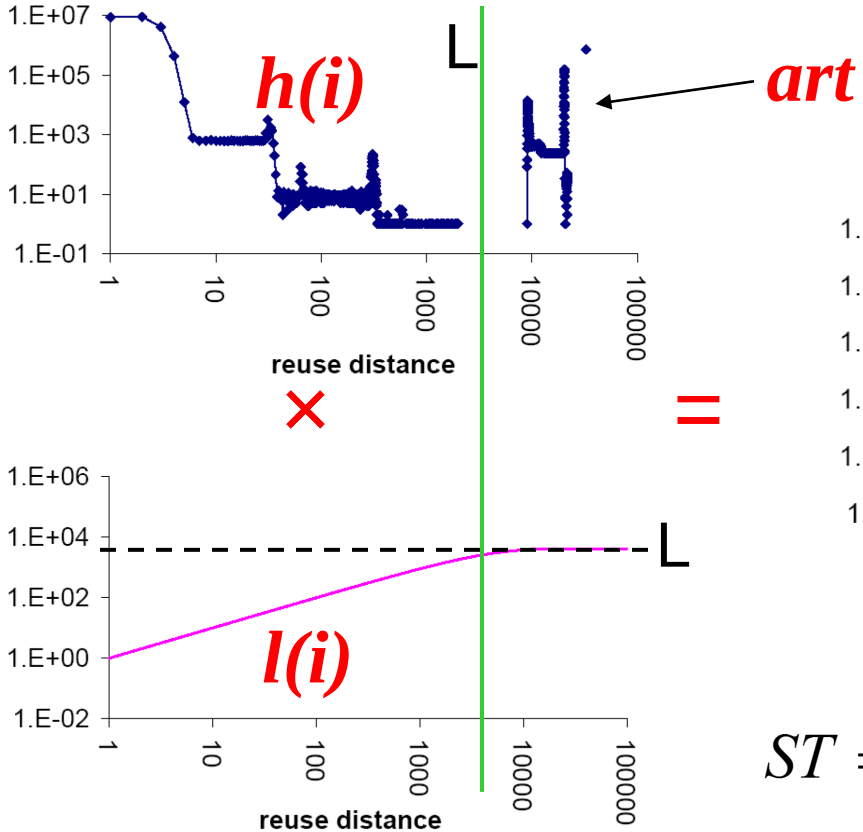
S: Spacetime

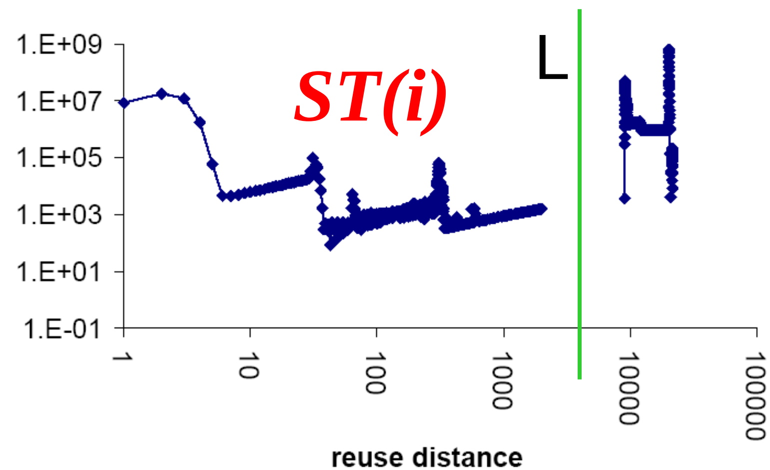
Space time

- Spacetime: “mean active ratio” (footprint) over a period of time
- Spacetime: cachelines x lifetime
- Relates occupied space to misses

Spacetime

- Lifetime expectancy: $l(i) = L \times f(i)$

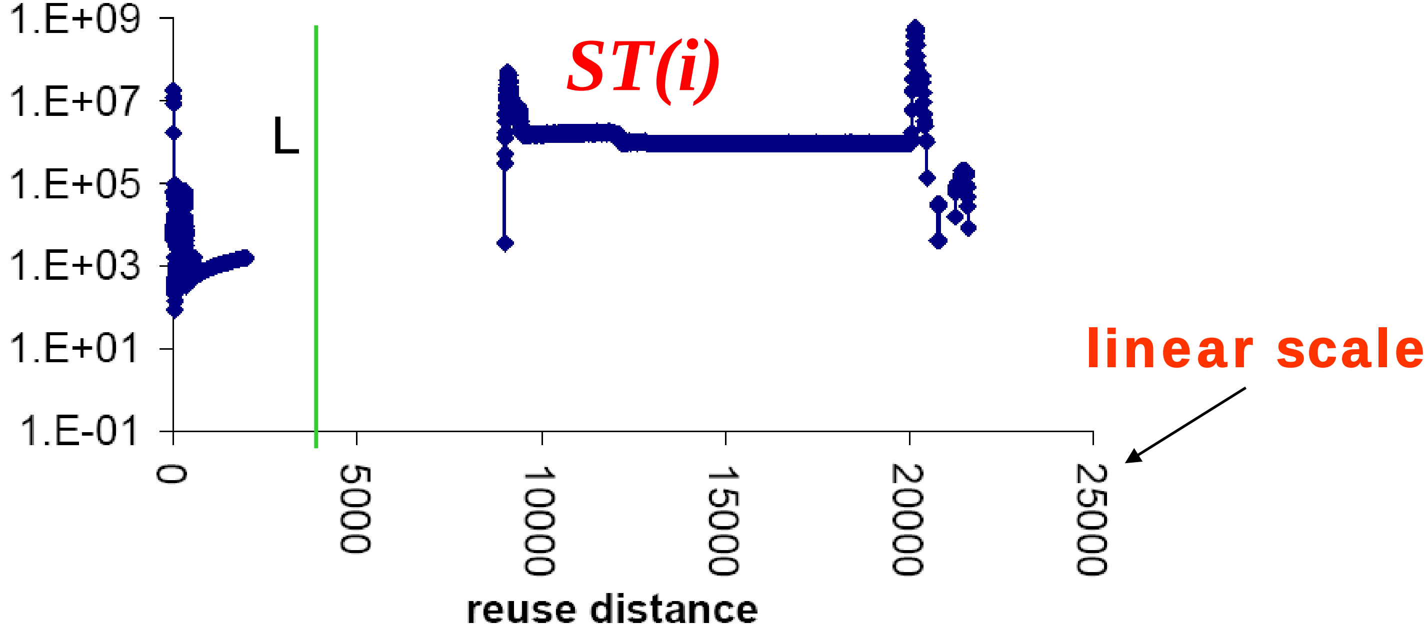




$$ST = \sum_{i=0}^{\infty} L \times f(i) \times h(i) = L \times misses$$

Arguing for management

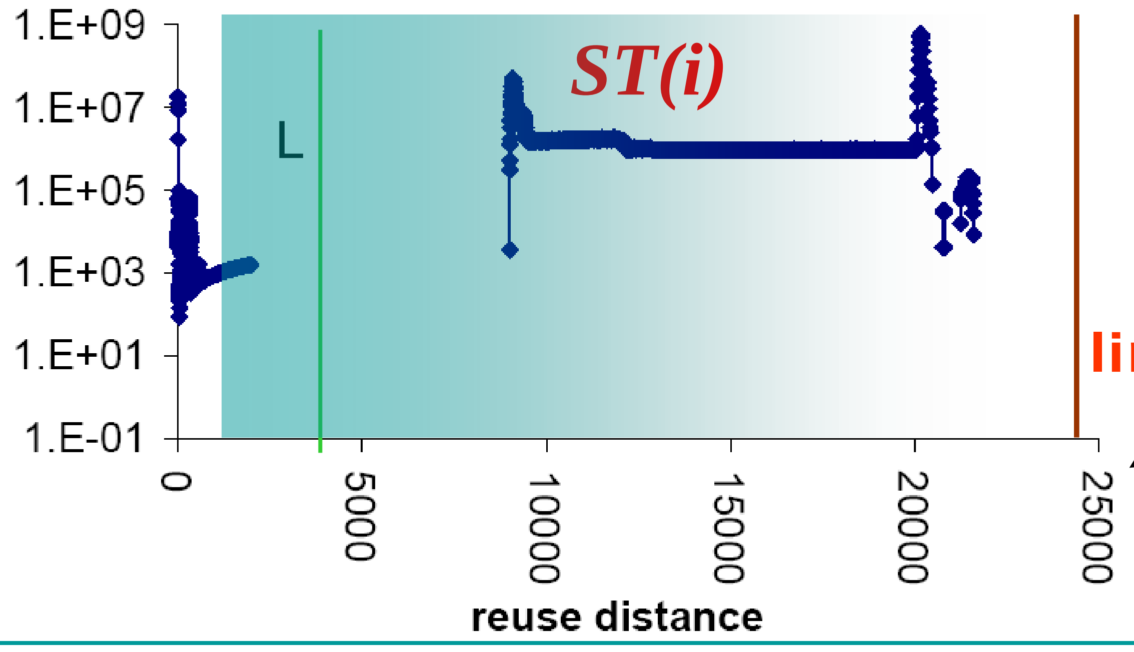
- Cachelines with long reuse distances
  - Have very low probability of producing a hit
  - Occupy most of the cache space



Arguing for management

- Cachelines with long reuse distances
  - **Should be constrained**

Decay!



linear scale

Original Cache Decay

- Leakage reduction in uniprocessor caches
- Lines not accessed over a period of time (Decay Interval) are turned off (decayed)
- Restricting the application to a small footprint
  - *With minimal performance loss*

Management via (a new form of) Decay

- *CAT decay*
- Decay intervals measured in *CAT not* cycles
  - Same notion of time allows integration into our model
- Not leakage reduction, but *replacement policy*
  - Decayed lines not really turned off, just available for replacement
  - Random replacement only when no decayed line is available

Management via (a new form of) Decay

- Active ratio control via CAT decay interval
  - Flexible, fine grain management
  - Not a strict policy: allows spikes and dips
  - Superior to strict partitioning (fine or coarse-grained)

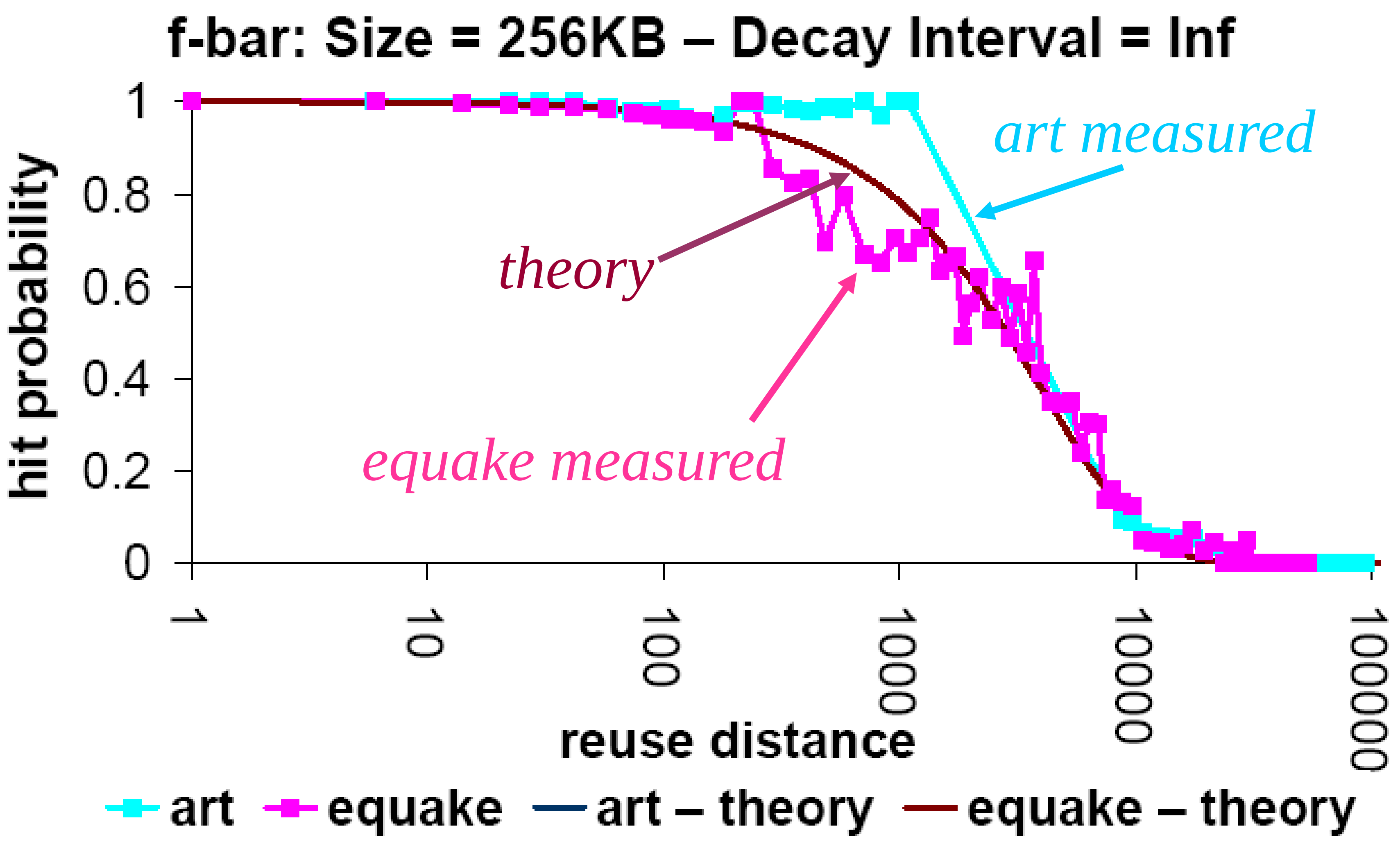
Decayed f-functions: introducing decay in the model

- Decay changes f-functions
  - Existence of decayed lines changes replacement policy
  - Not random anymore!
- Many different approximations possible ...
- MURPHY vs. managed replacements
  - Managed: finding a decayed line to replace
  - Murphy: using the underlying replacement policy (RR)
- Probability Murphy: $\mu = \dfrac{Murphy\_replacements}{Total\_replacements}$

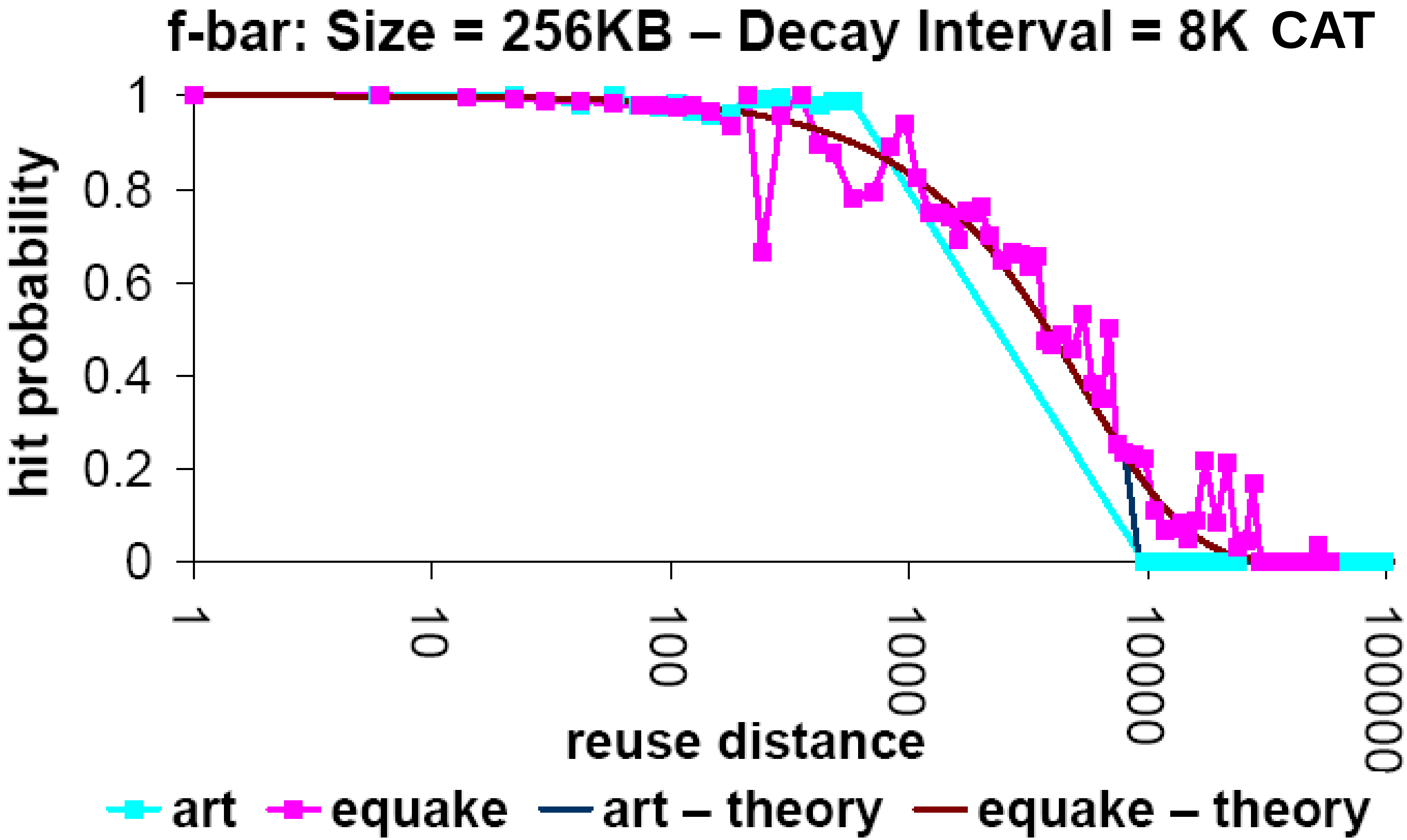
Decayed f-functions: introducing decay in the model

- How the f-functions change?
- For reuse-distances less than the Decay Interval:
  - ***L* is replaced by *L/μ***
  - Non-decayed applications "think" they share a much larger cache!
- For reuse-distances larger than the Decay Interval:
  - ***f* = 1 & *f-bar* = 0**

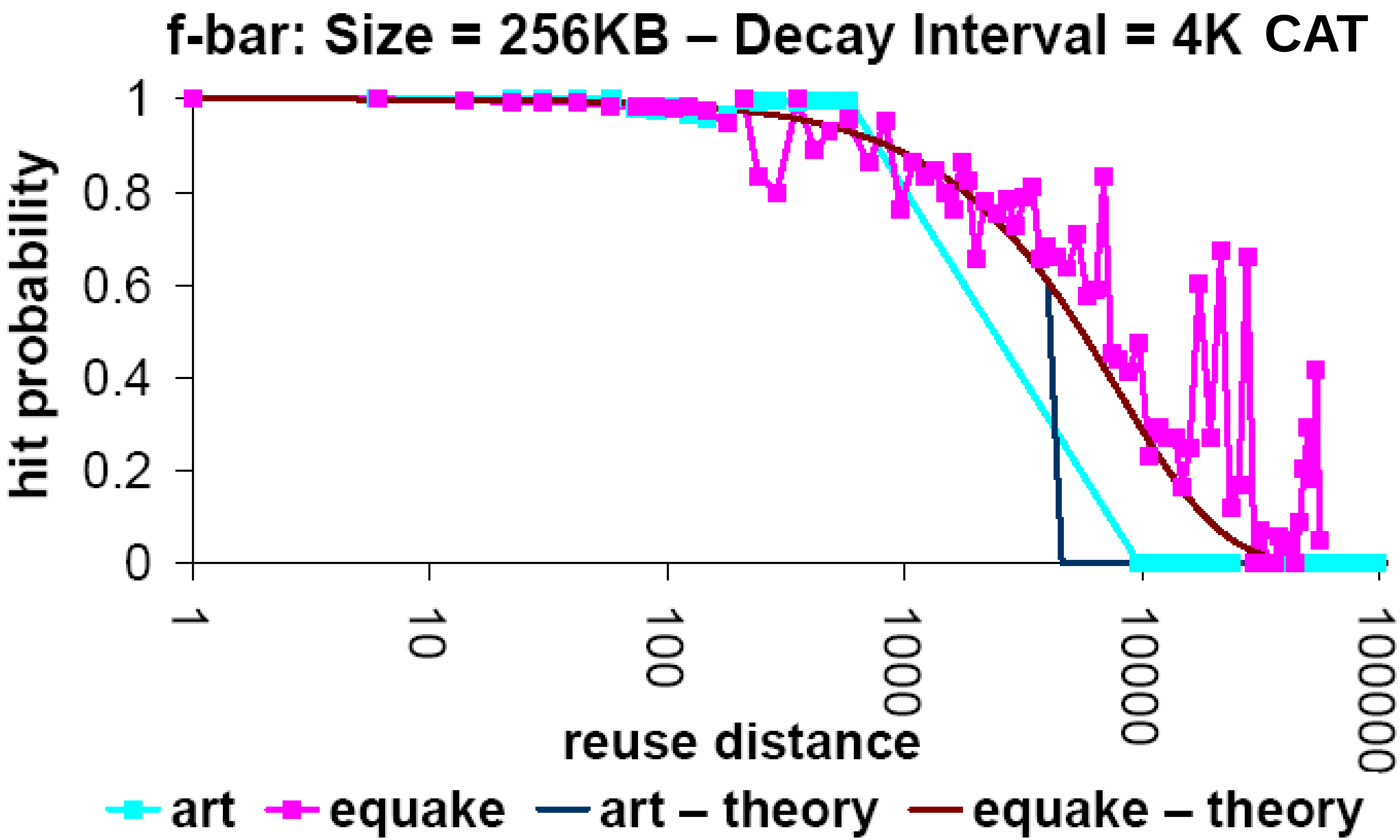
Non-Decayed f-bar functions



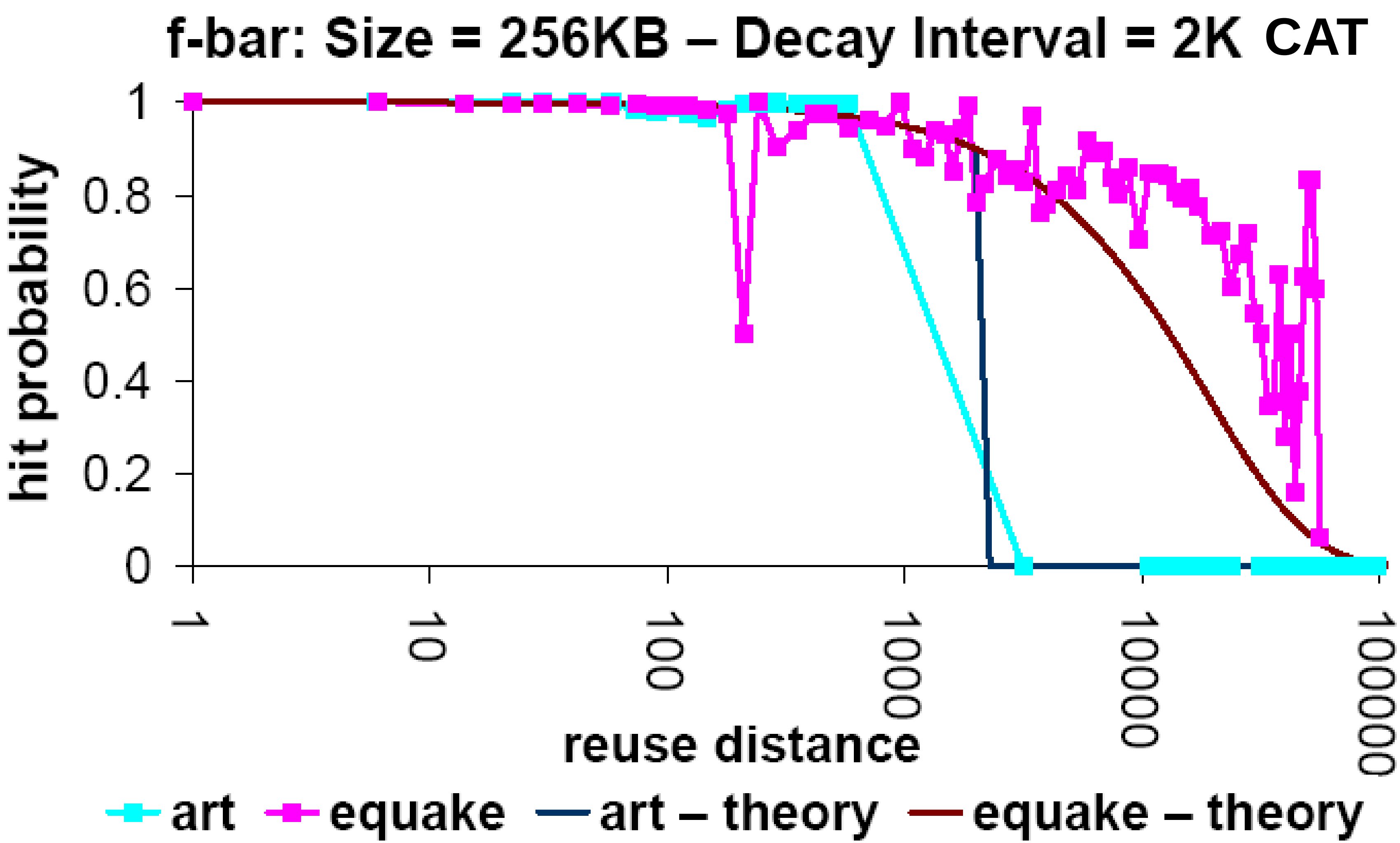
Decayed f-bar functions



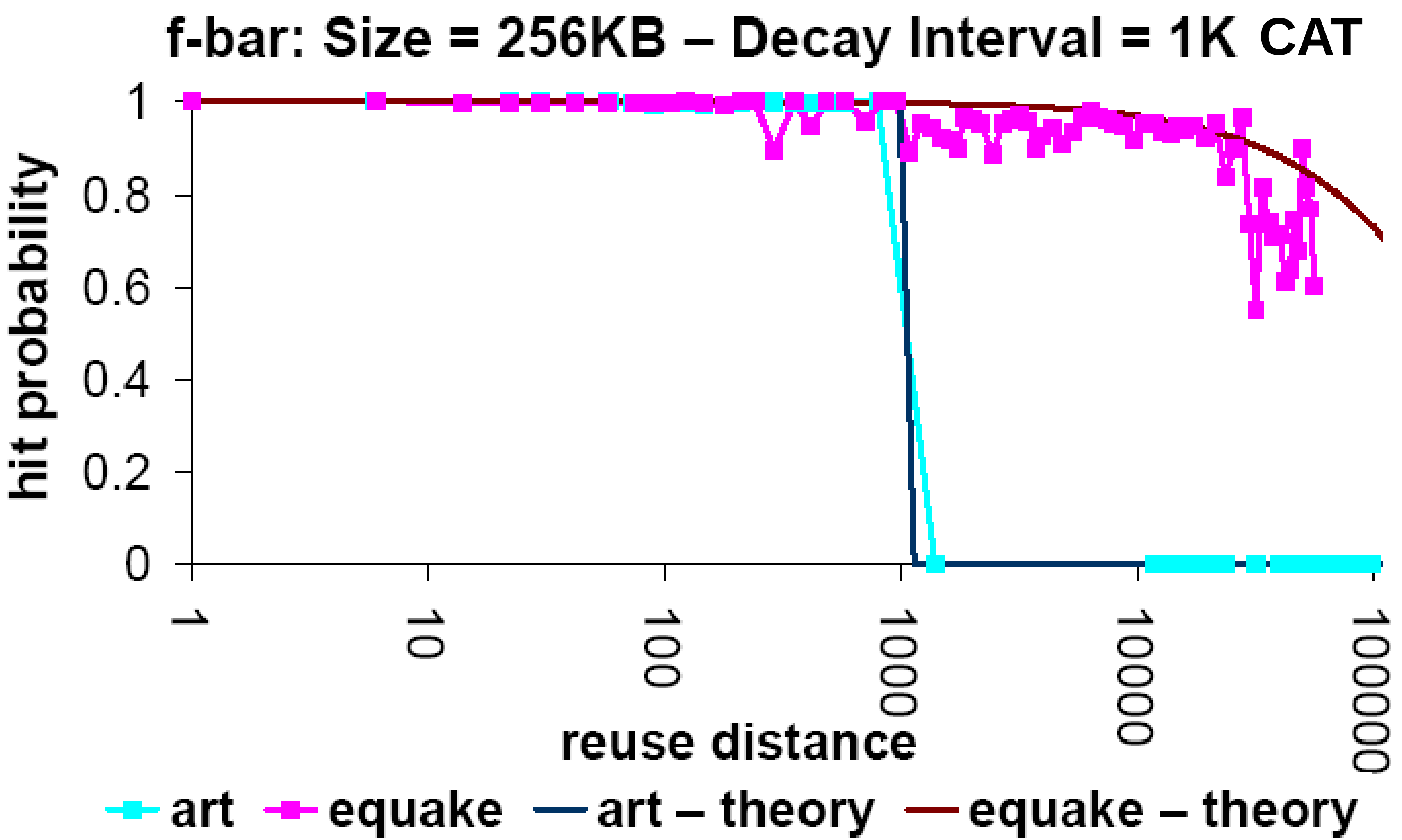
Decayed f-bar functions



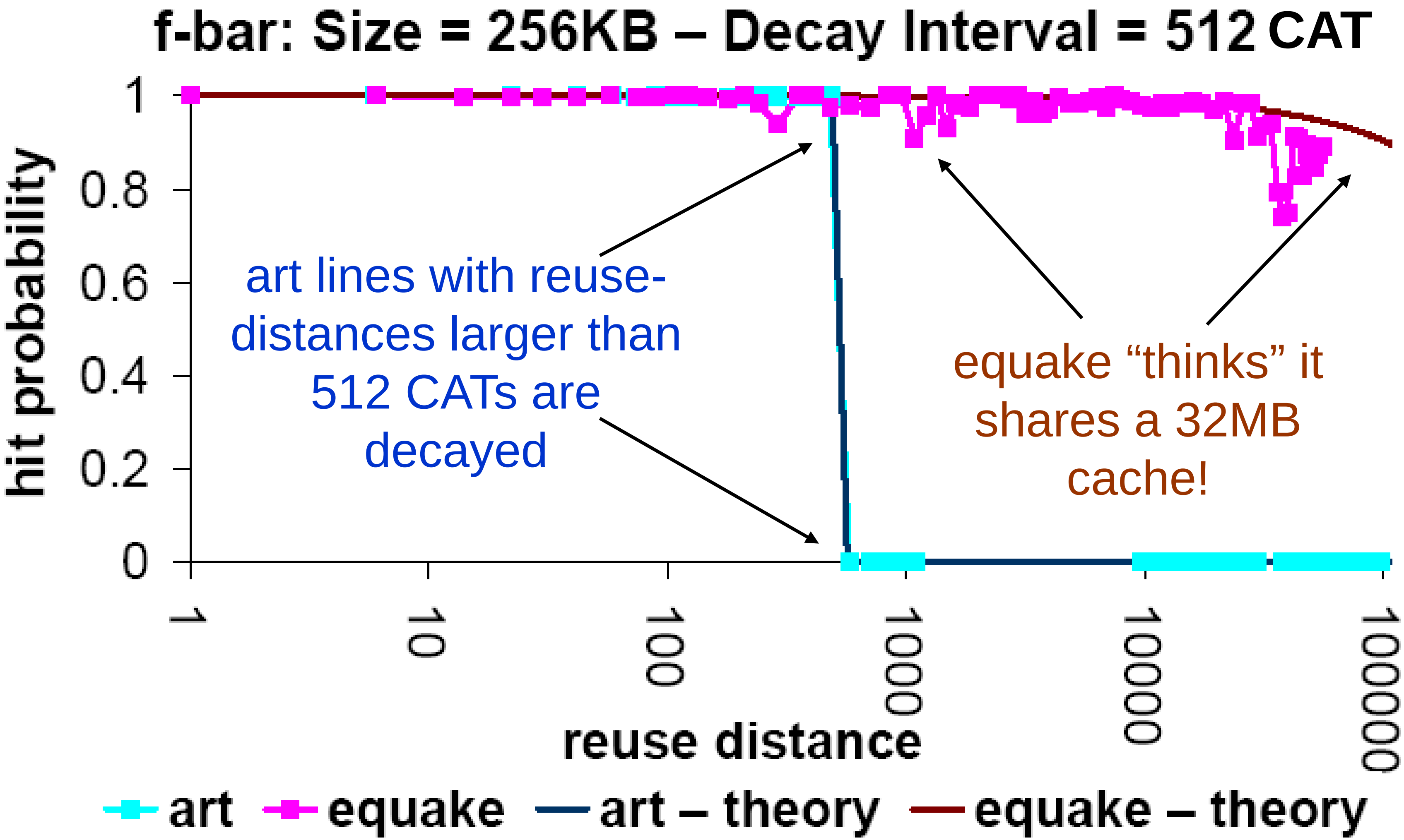
Decayed f-bar functions



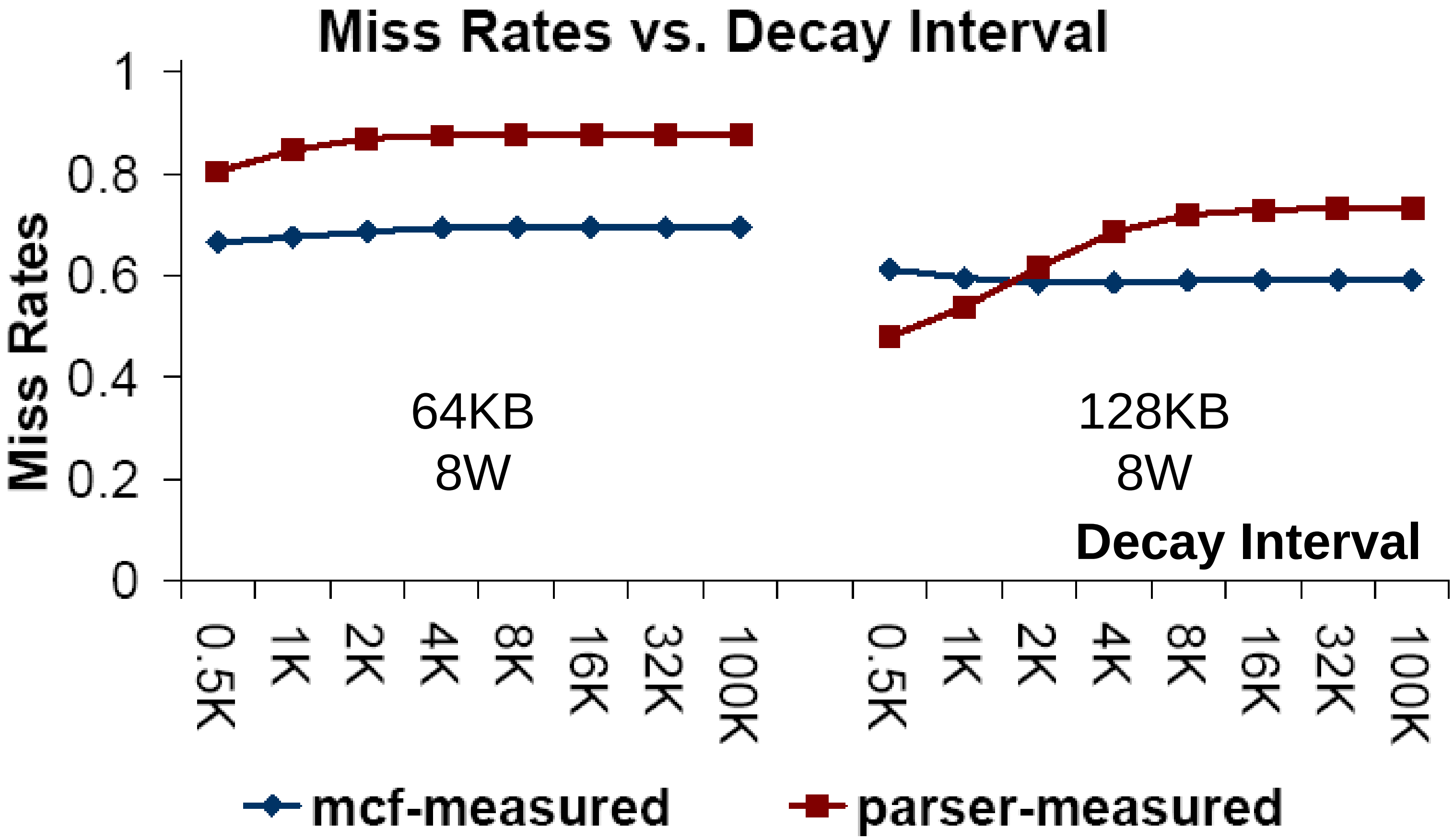
Decayed f-bar functions



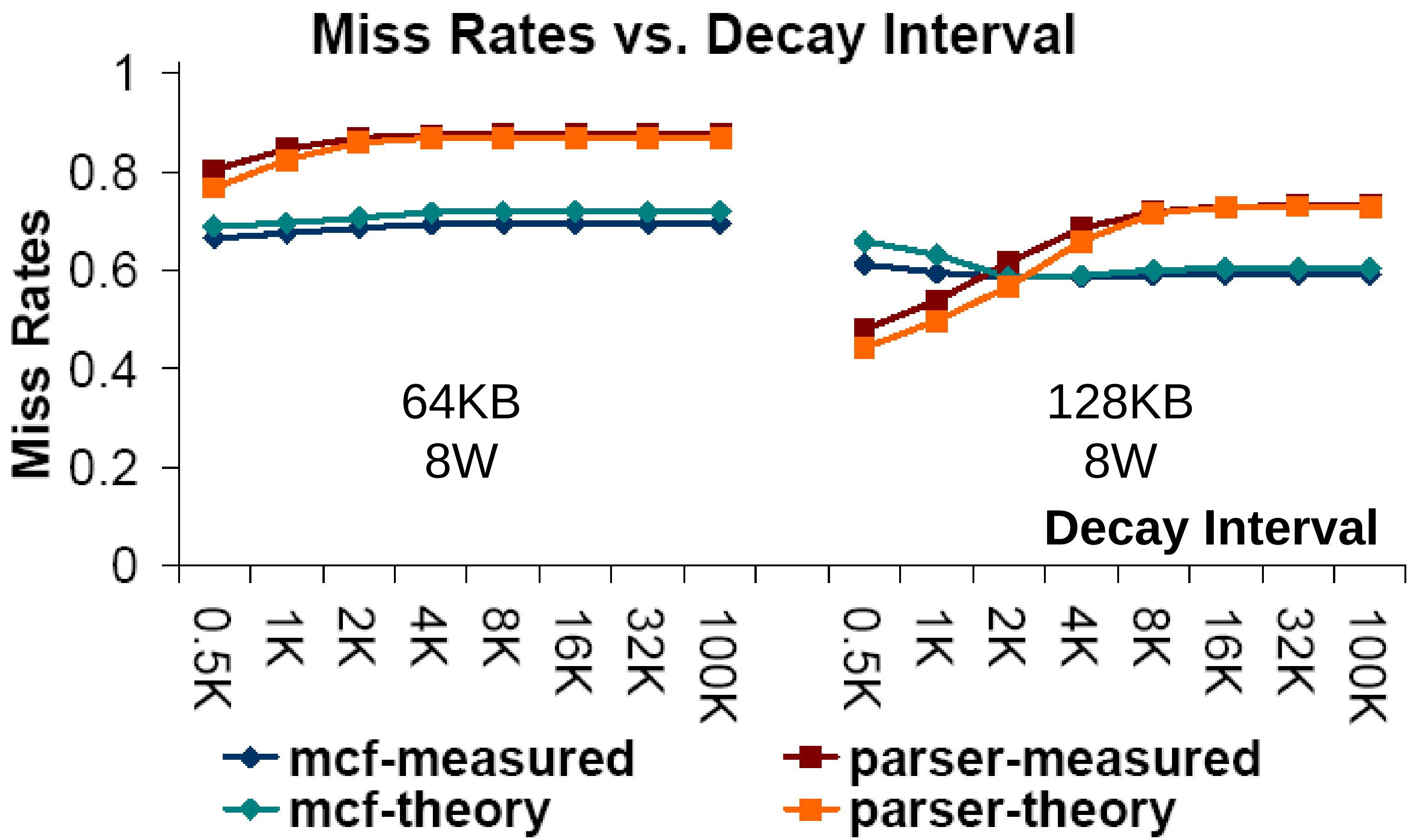
Decayed f-bar functions



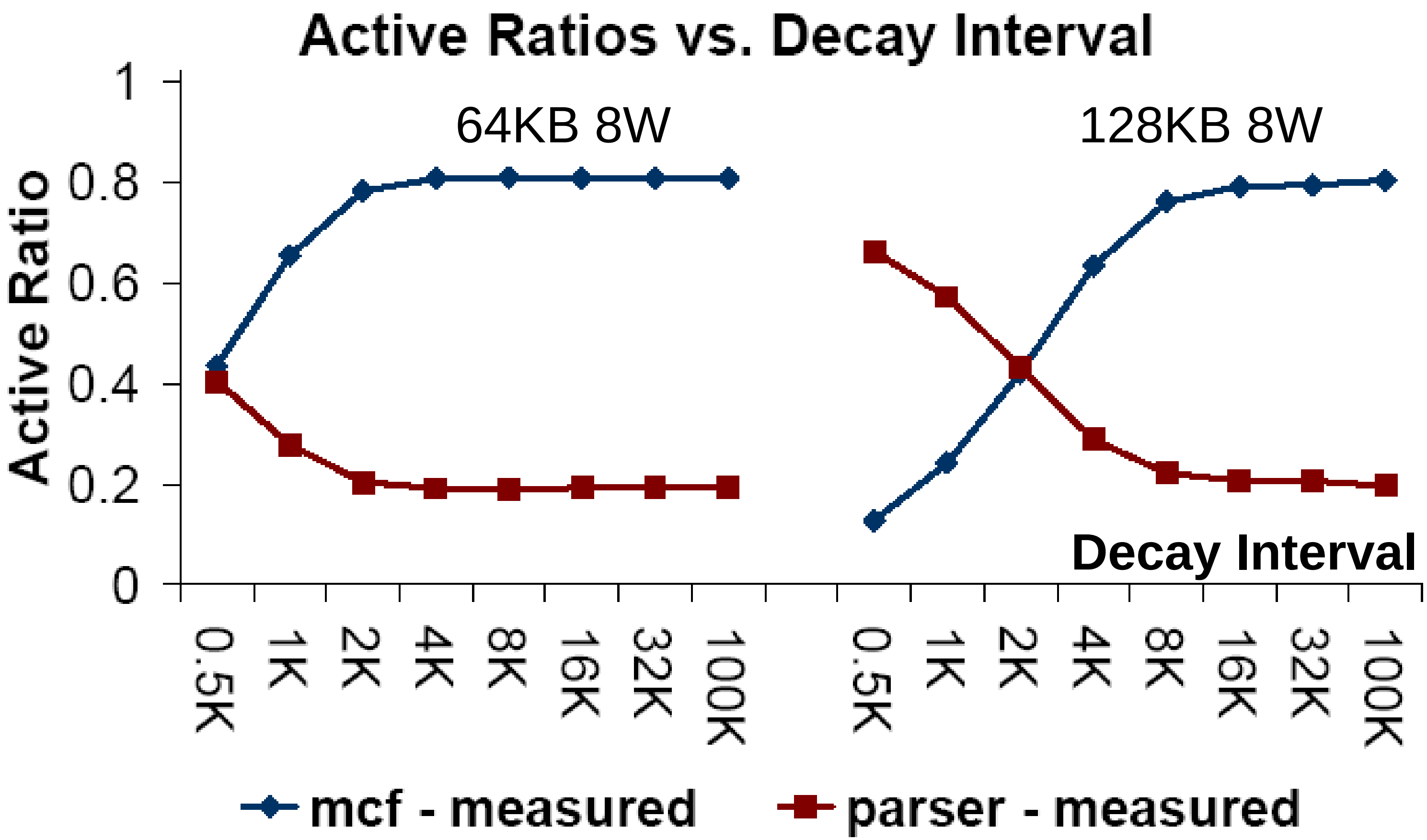
Model validation: mcf-parser – Miss Rates



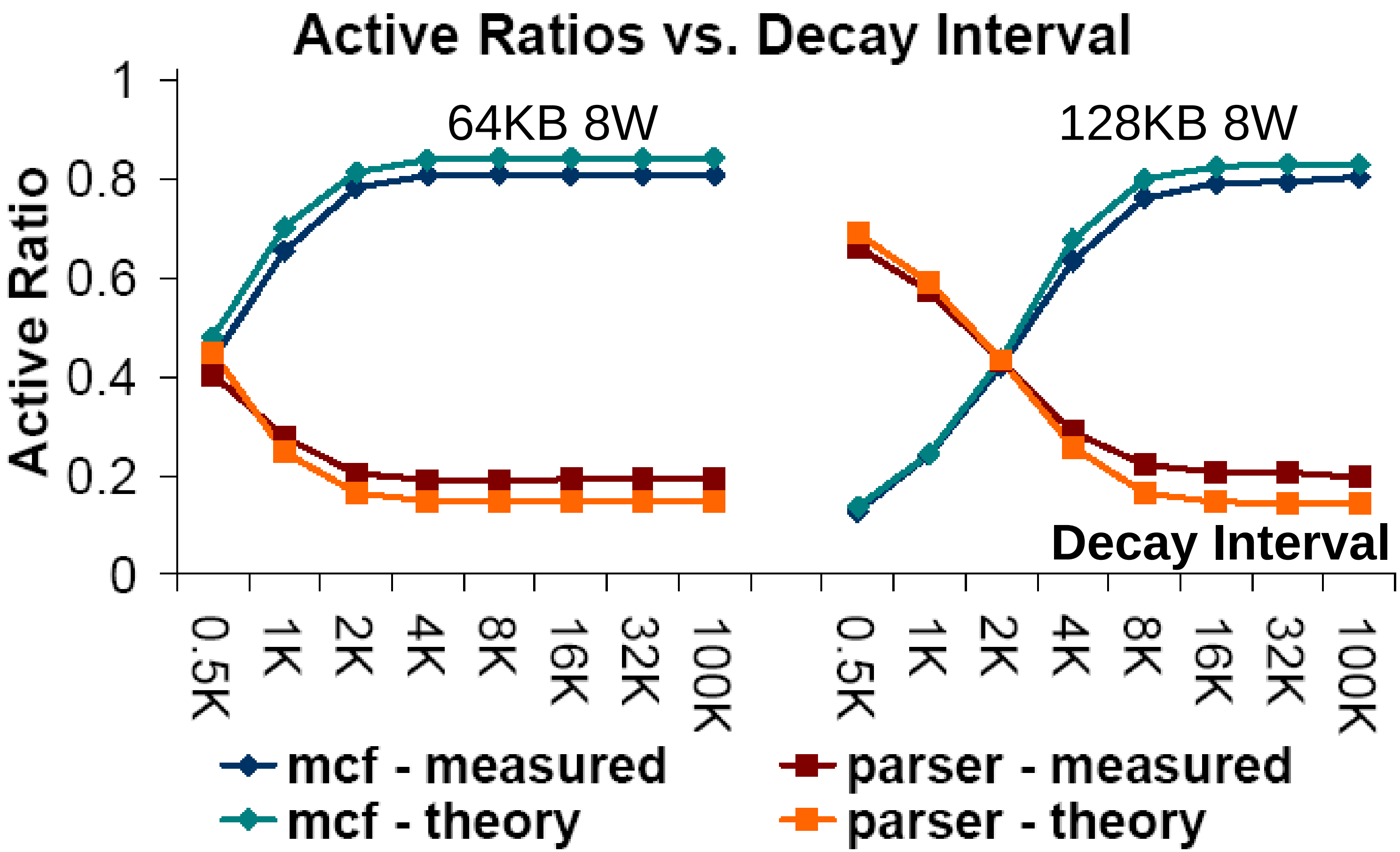
Model validation: mcf-parser – Miss Rates



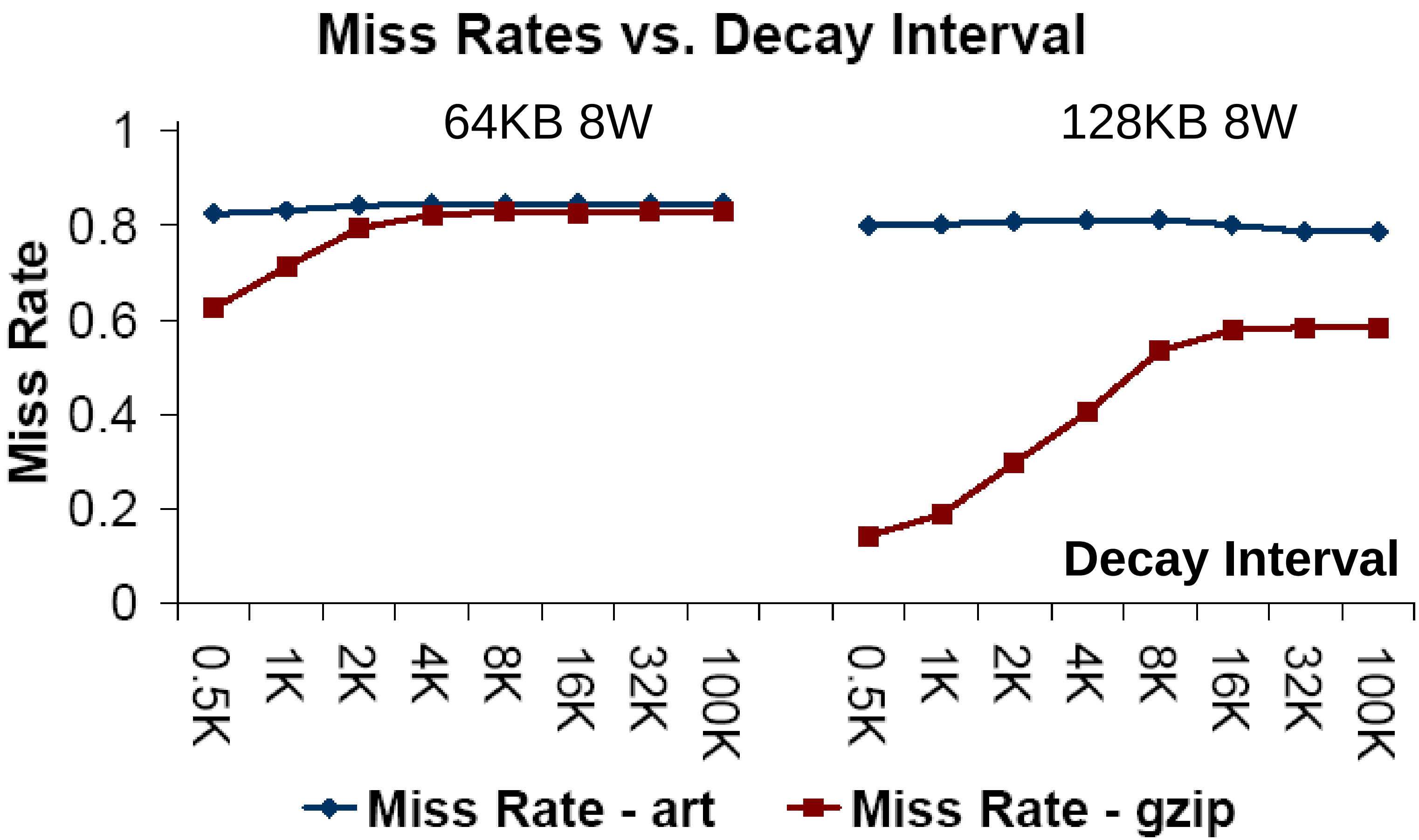
Model validation: mcf-parser – Active Ratios



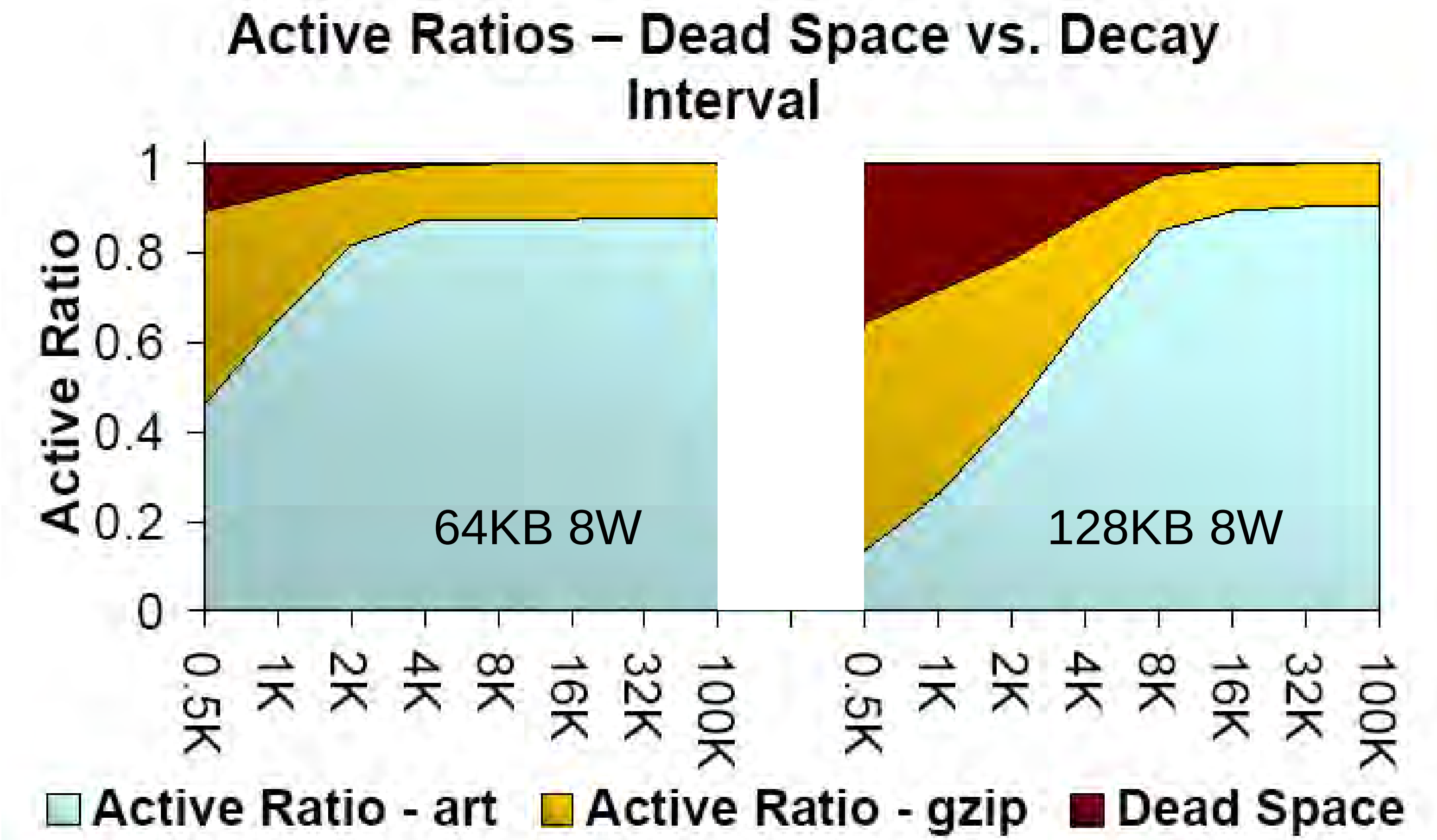
Model validation: mcf-parser – Active Ratios



Reasoning about cache management: art-gzip – Miss Rates



Reasoning about cache management: art-gzip – Active Ratios



More results in the paper…

- Model Validation with most memory intensive SPEC2000
  - 2 threads
  - 4 threads
- Cache Management:
  - 2 threads
    - 1 thread decayed
  - 4 threads
    - 1 – 4 threads decayed

Conclusions

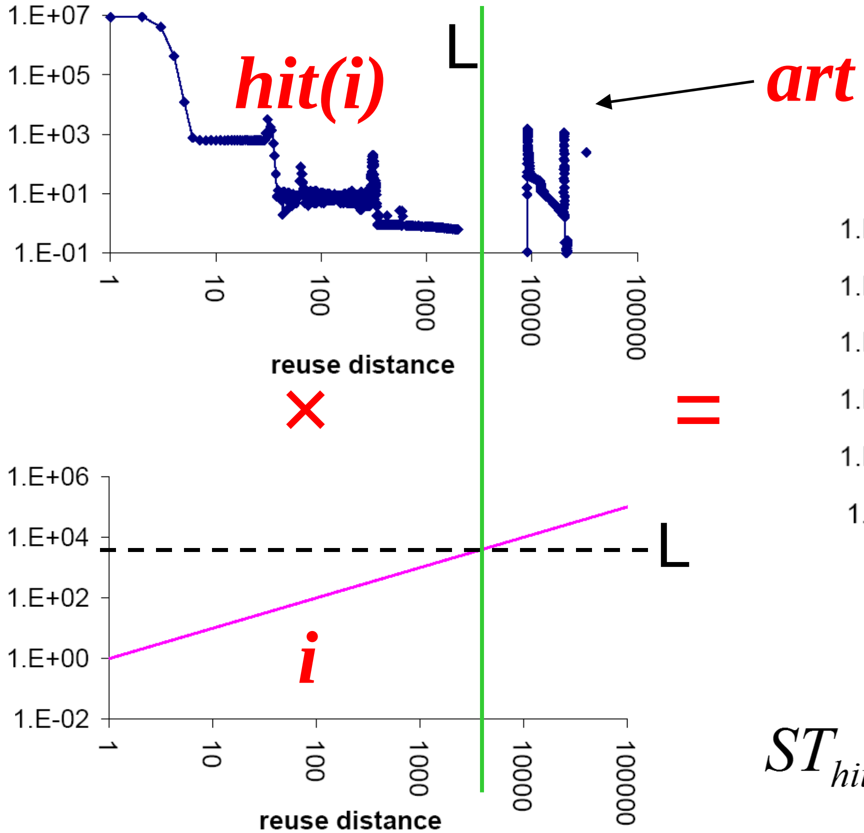
- Model:
  - Hits, misses, active ratios...
  - Can be computed @ runtime
- Control Mechanisms
  - CAT Decay
  - Other cache partitioning approaches...
- Use such info to implement high-level policies

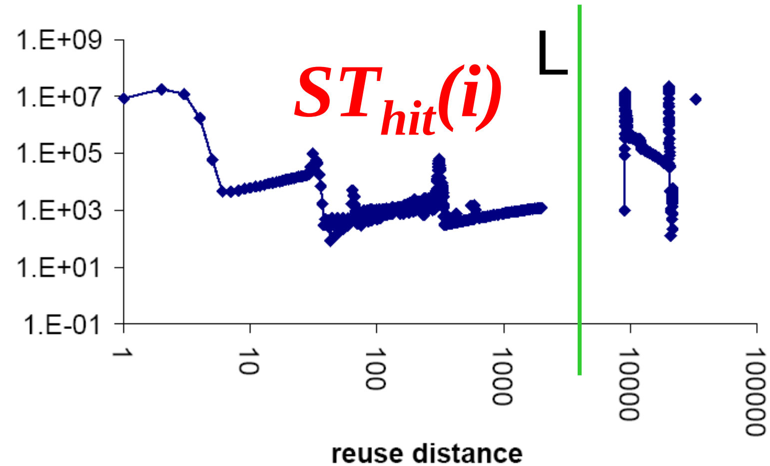
Questions

?

Hits Spacetime

- Hits occupy ST for their reuse distance

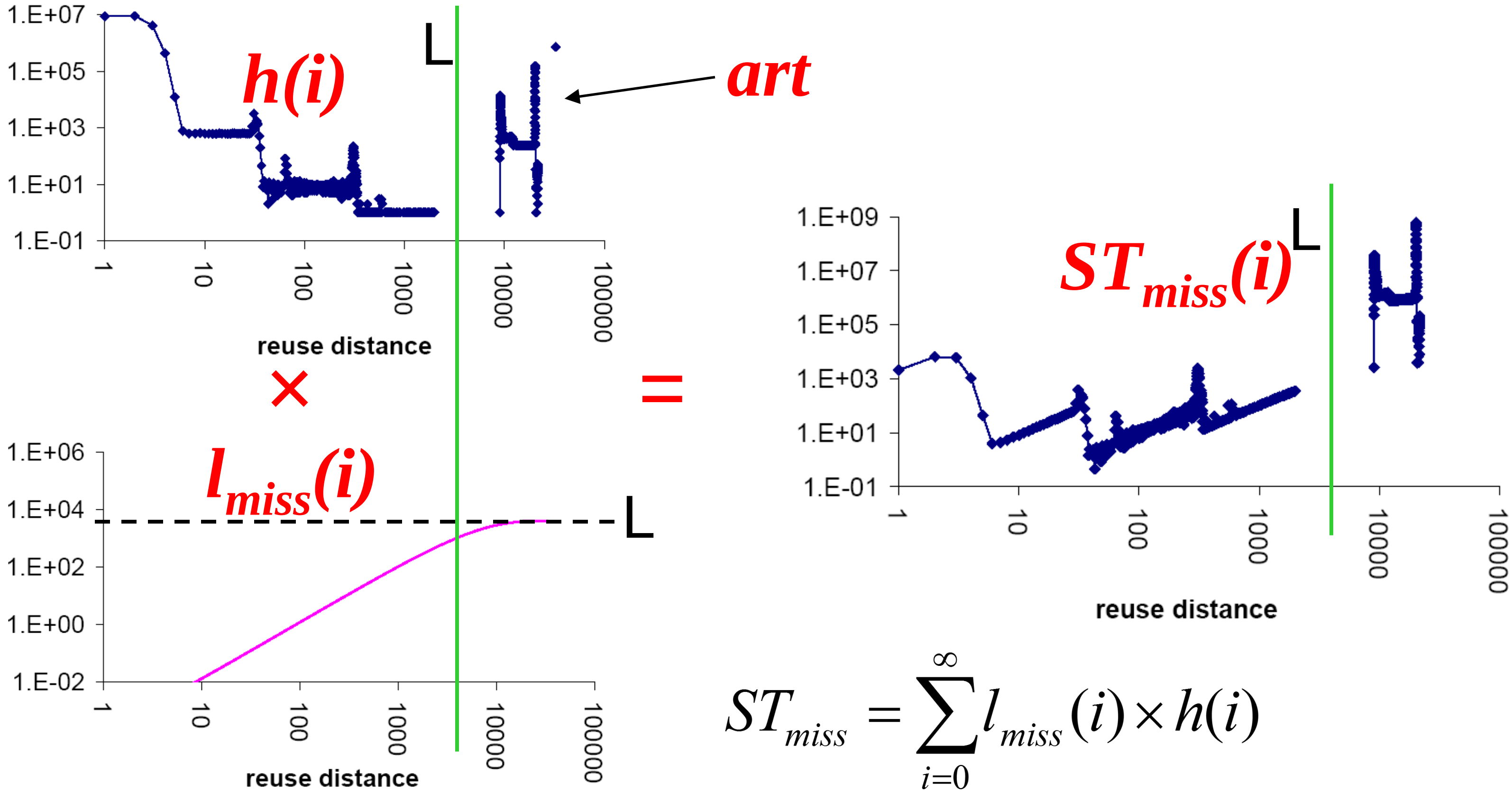




$$ST_{hit} = \sum_{i=0}^{\infty} i \times hit(i) = \sum_{i=0}^{\infty} i \times h(i) \times \overline{f}(i)$$

Misses Spacetime

- Misses occupy ST until their replacement



$$ST_{miss} = \sum_{i=0}^{\infty} l_{miss}(i) \times h(i)$$