

# Modeling Cache Sharing on Chip Multiprocessor Architectures

Pavlos Petoumenos<sup>1</sup>, Georgios Keramidas<sup>1</sup>, Håkan Zeffe<sup>2</sup>, Stefanos Kaxiras<sup>1</sup>, Erik Hagersten<sup>2</sup>

<sup>1</sup>University of Patras, Greece

<sup>2</sup>Uppsala University, Sweden



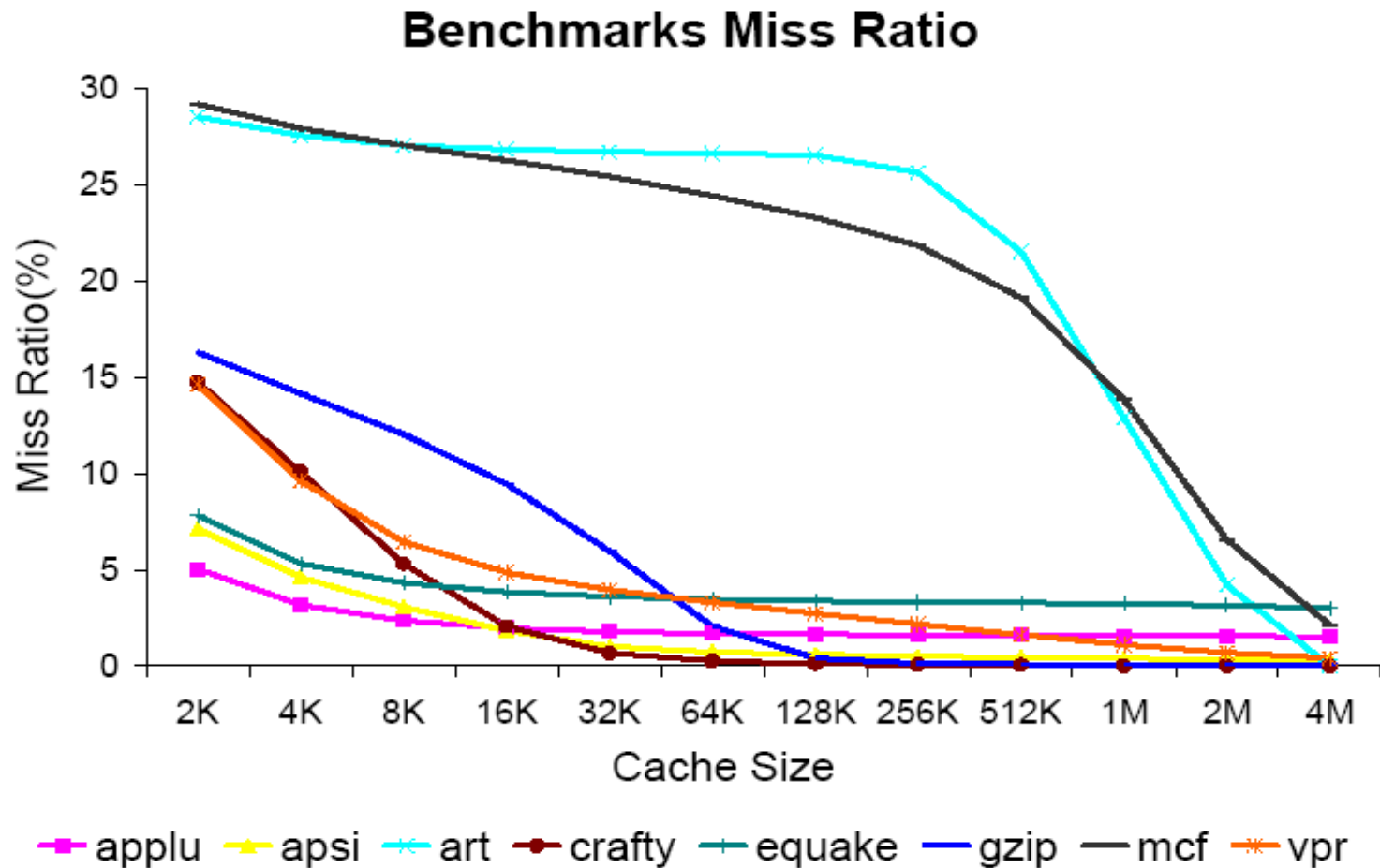
# Introduction

- 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**

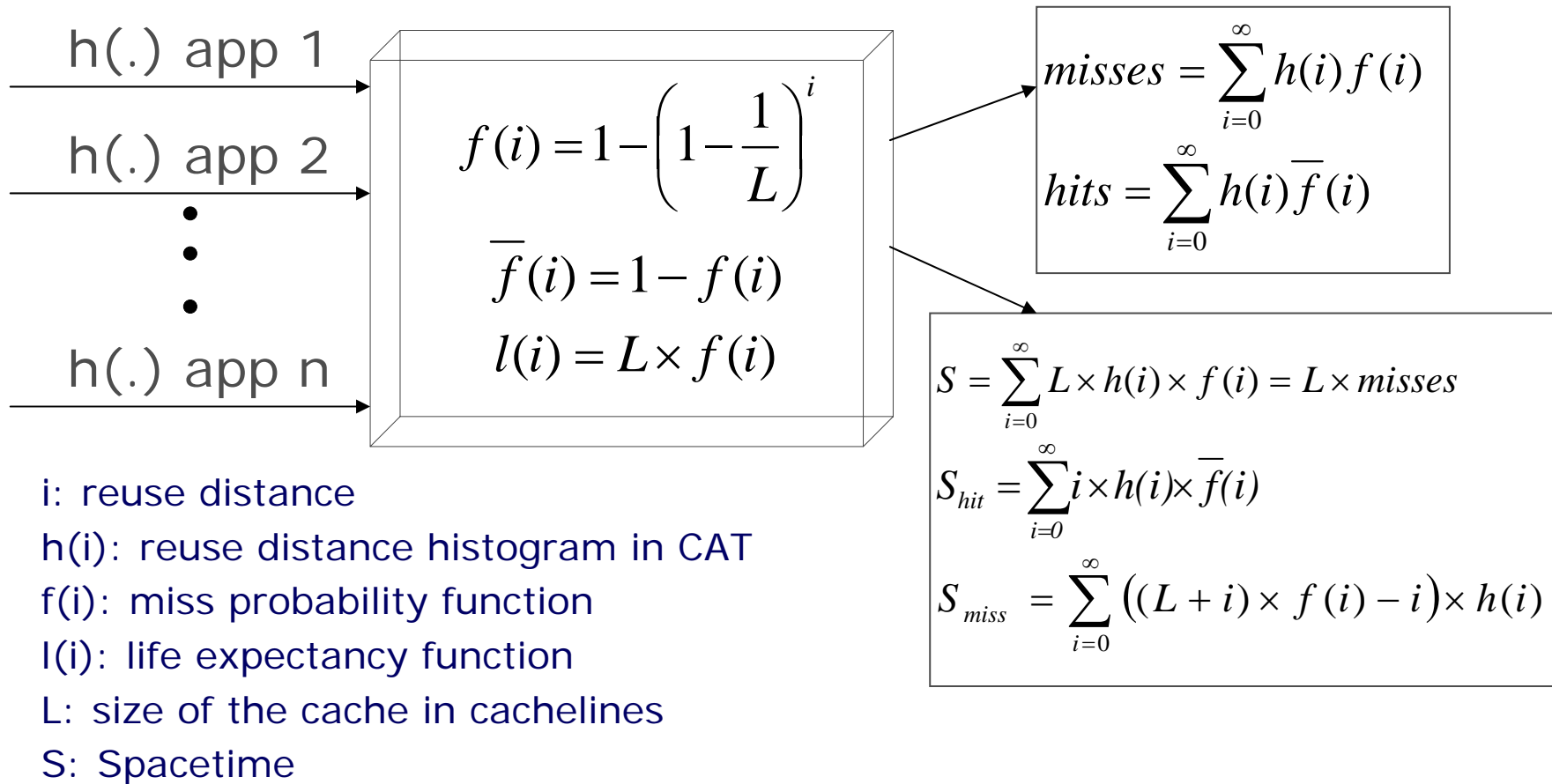
# Statcache output (motivation)



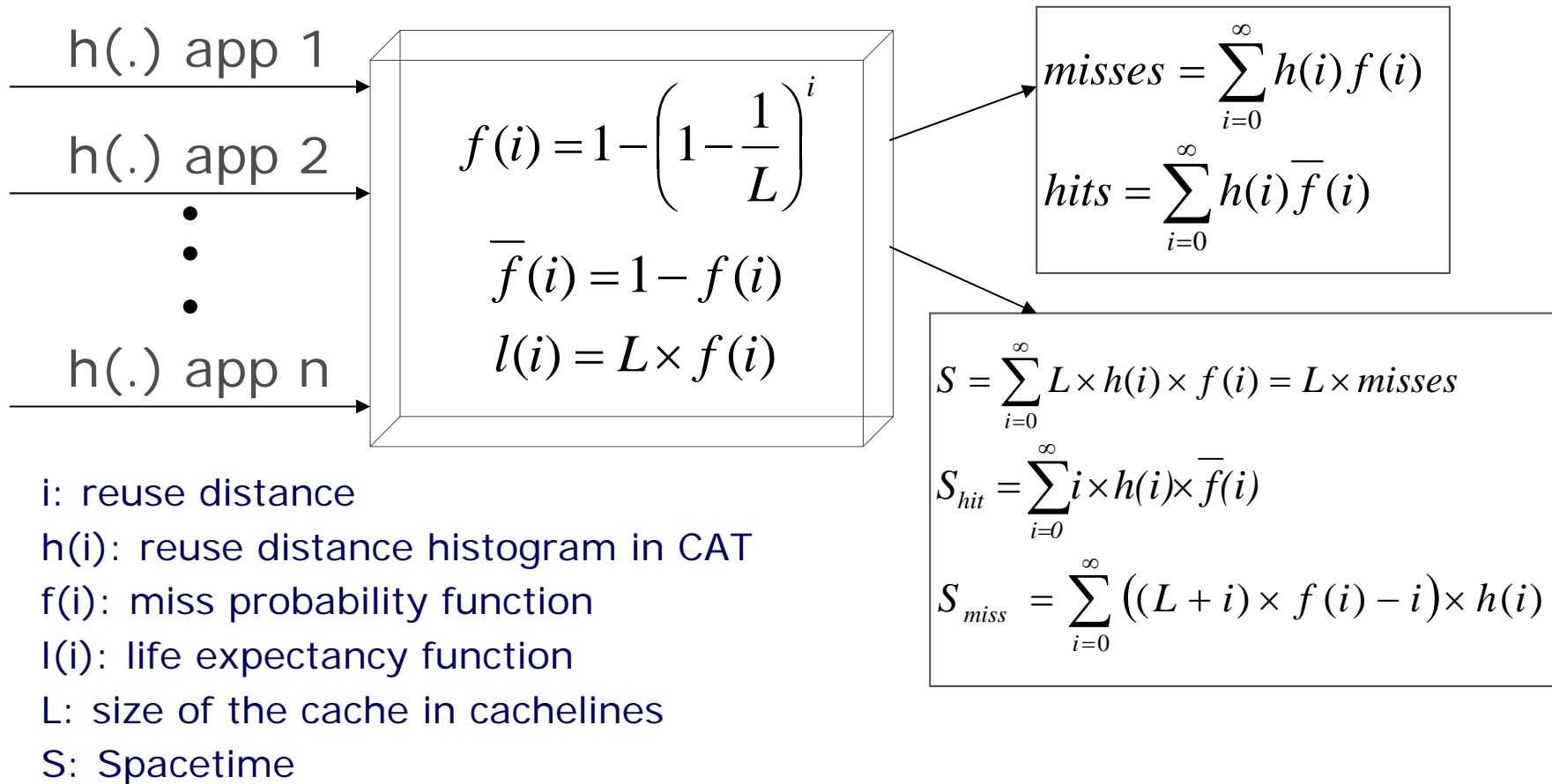
# StatShare

- 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  $\Rightarrow$  online model

# StatShare model

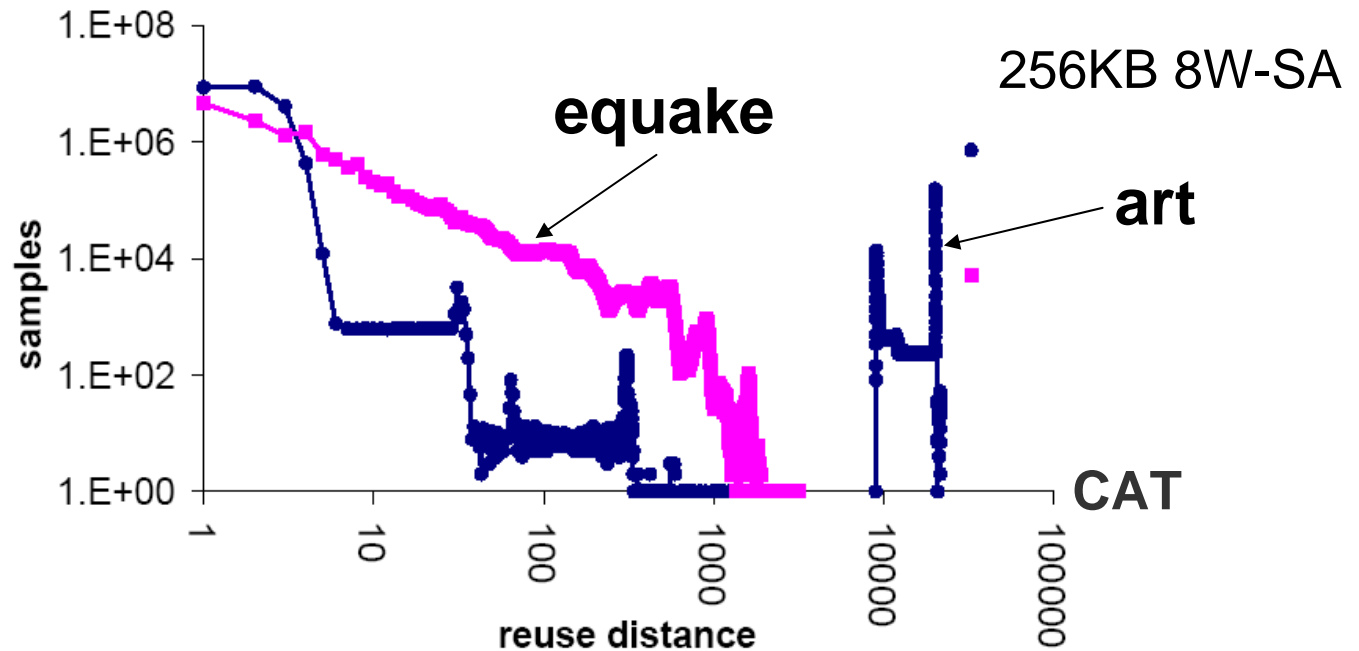


# StatShare model - Histograms



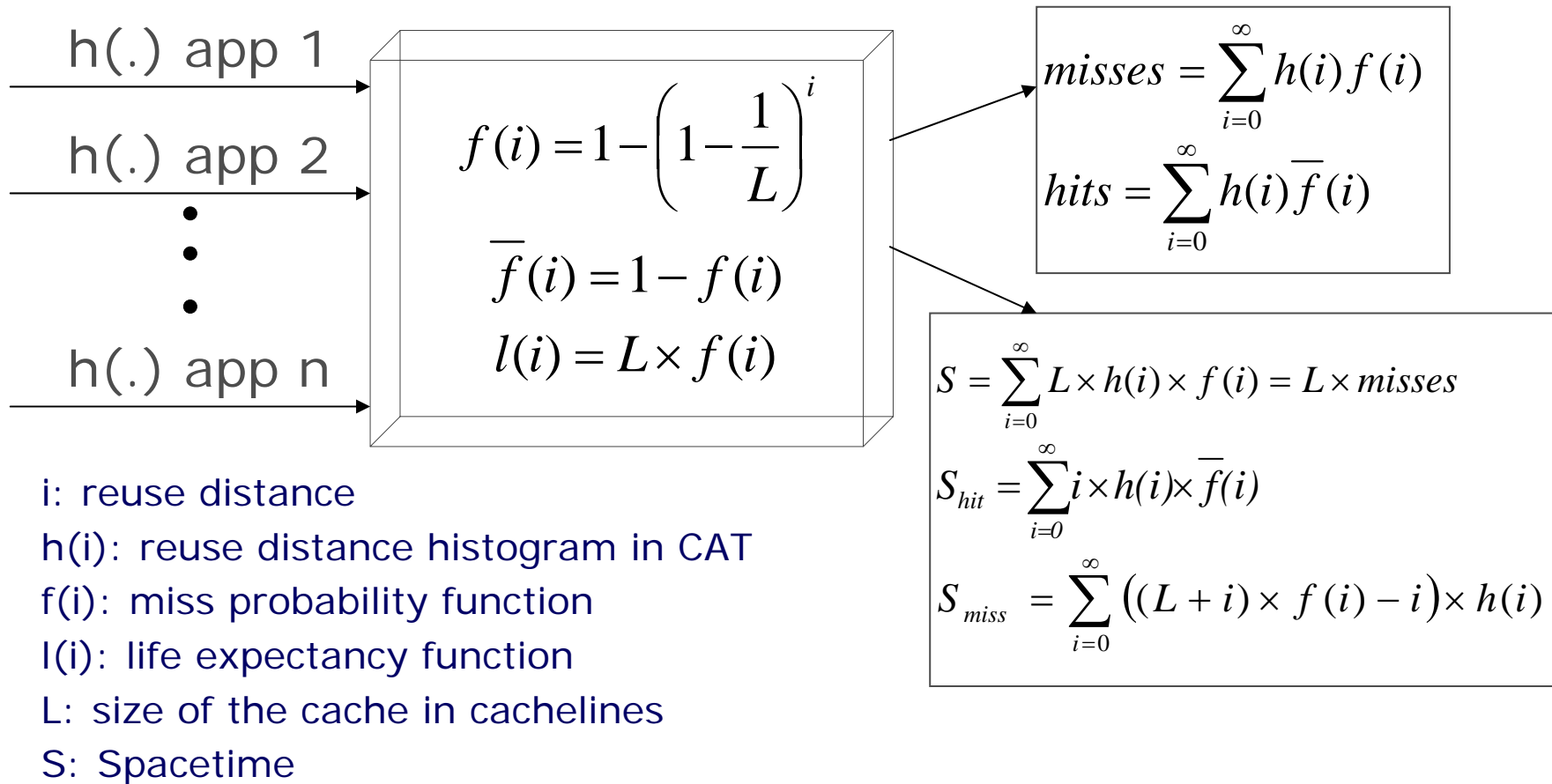
# CAT reuse distance histograms

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





# StatShare model – f-functions



# 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:  $(1 - 1/L)^i = \mathbf{f\text{-}bar}$

# 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:  $1/L$

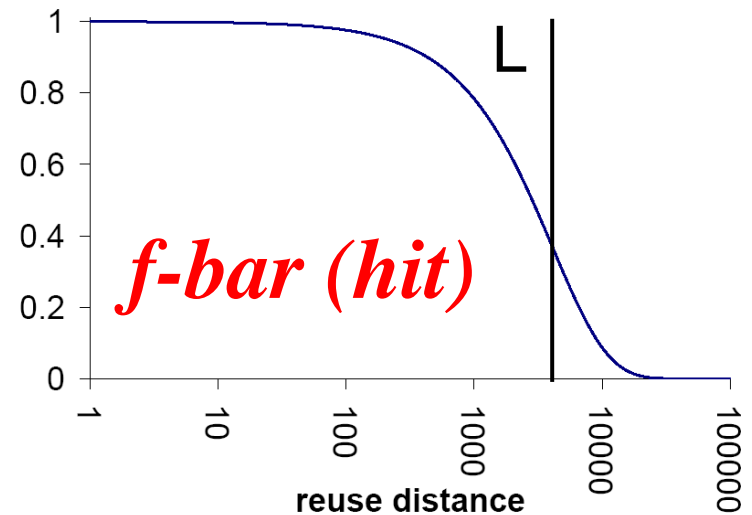
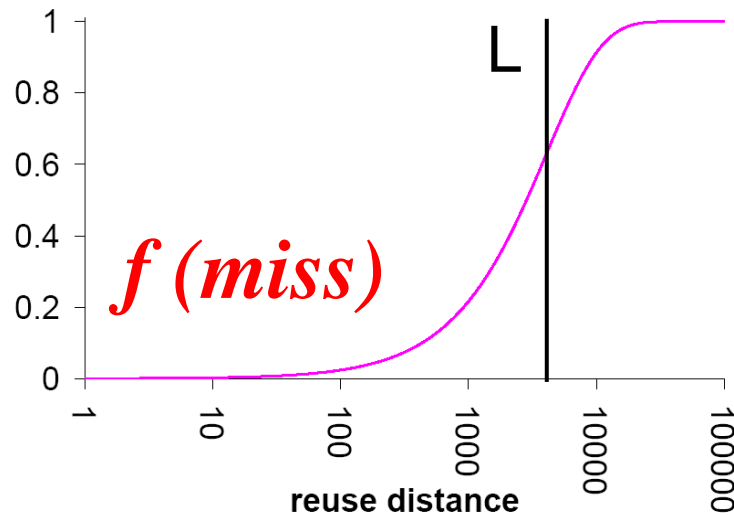
**Not** being selected for replacement:  $1 - 1/L$

Remaining in the cache after  $i$  CAT:  $(1 - 1/L)^i = \mathbf{f\text{-}bar}$

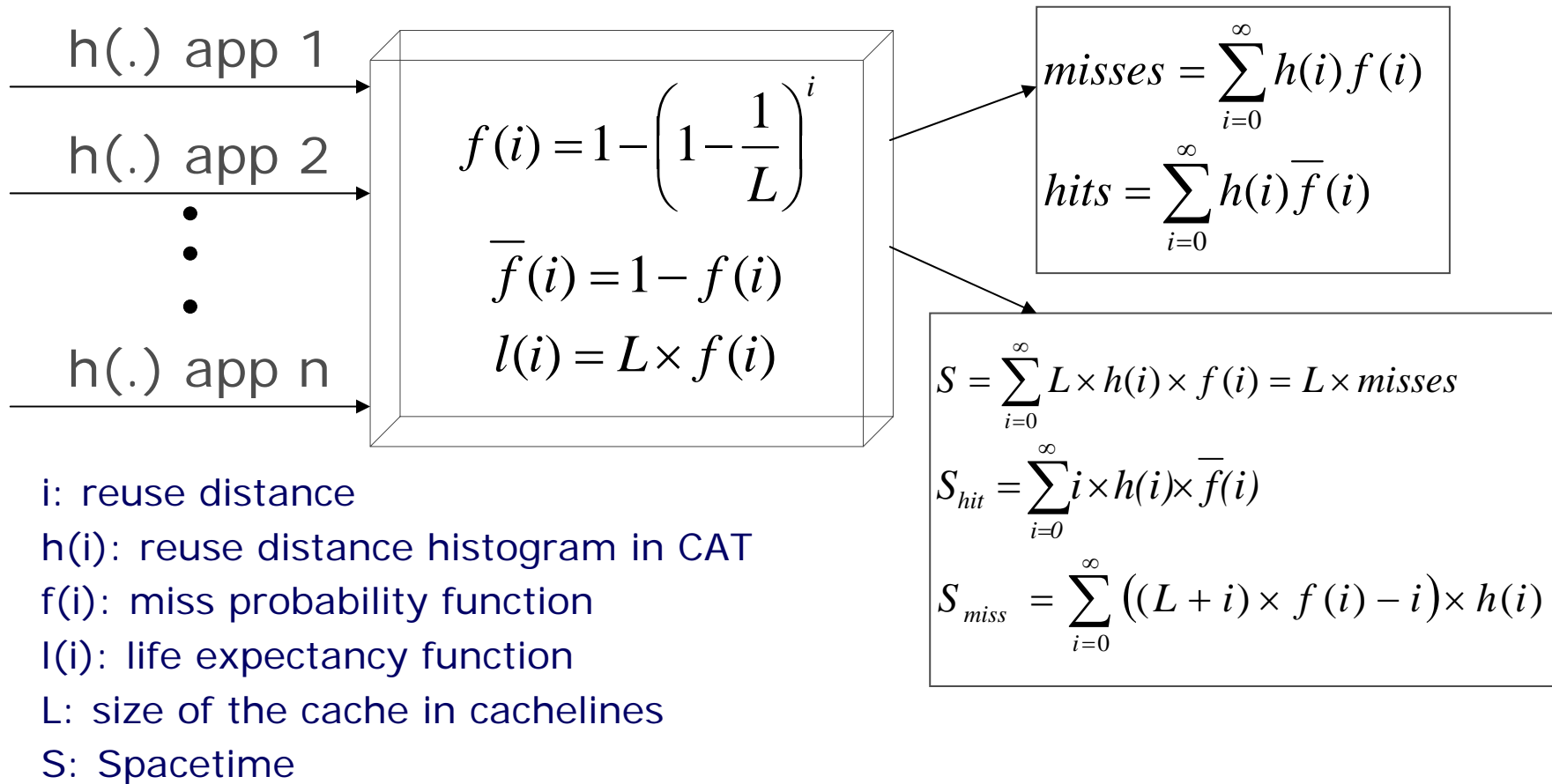
Having been replaced after  $i$  CAT:  $1 - (1 - 1/L)^i = \mathbf{f}$

# 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



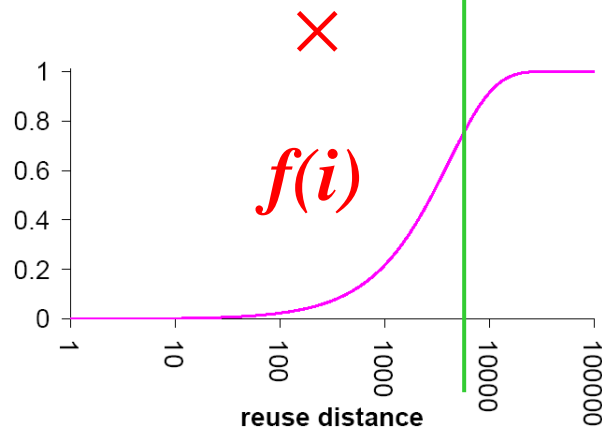
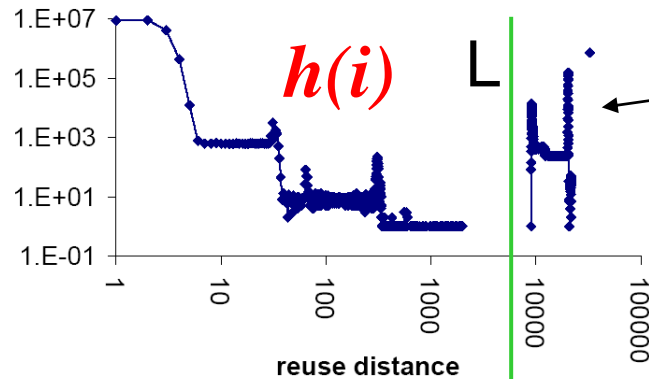
# StatShare model – hits & misses



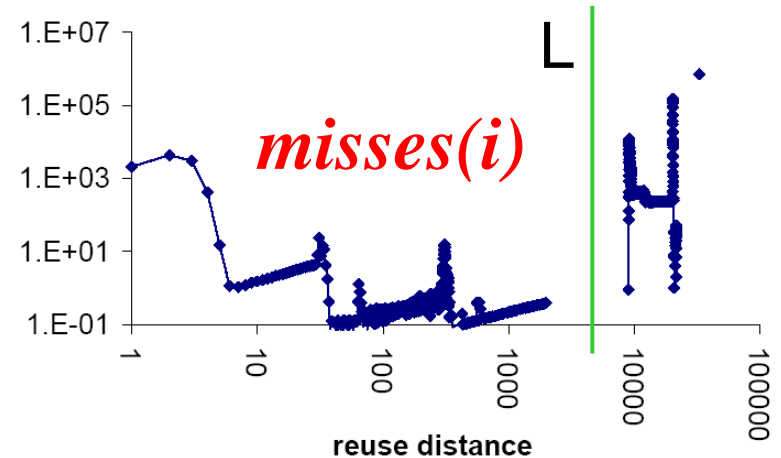


# misses

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



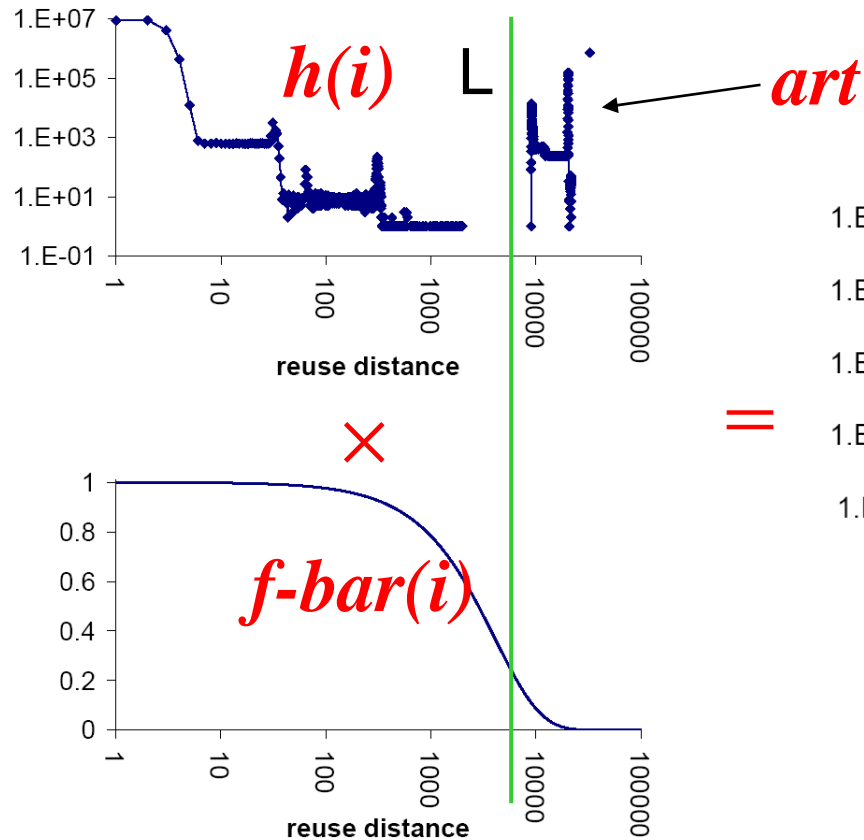
=



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

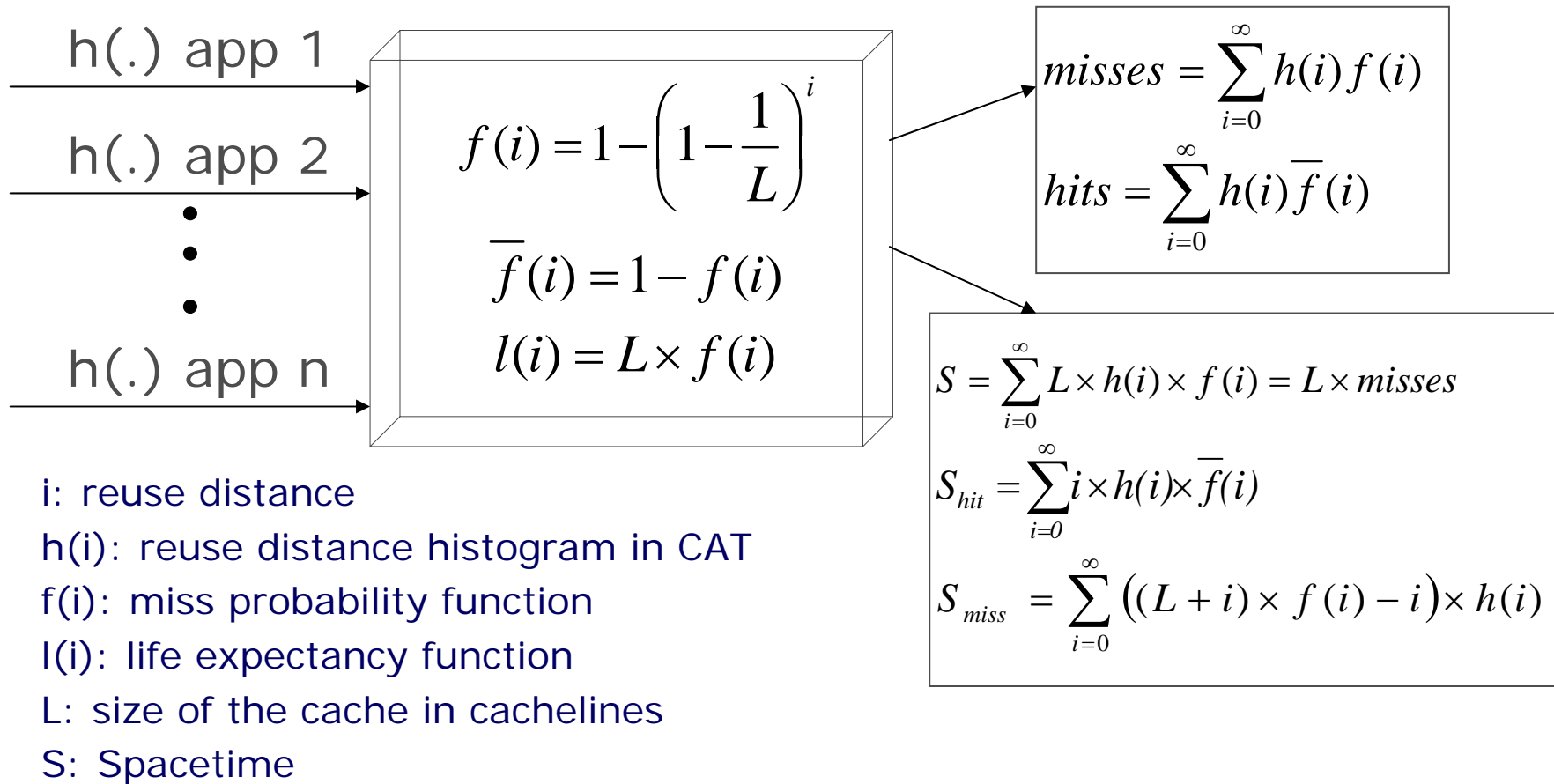
# hits

- $h$  multiplied by  $f\text{-bar}$  gives us hits



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

# StatShare model – Space time

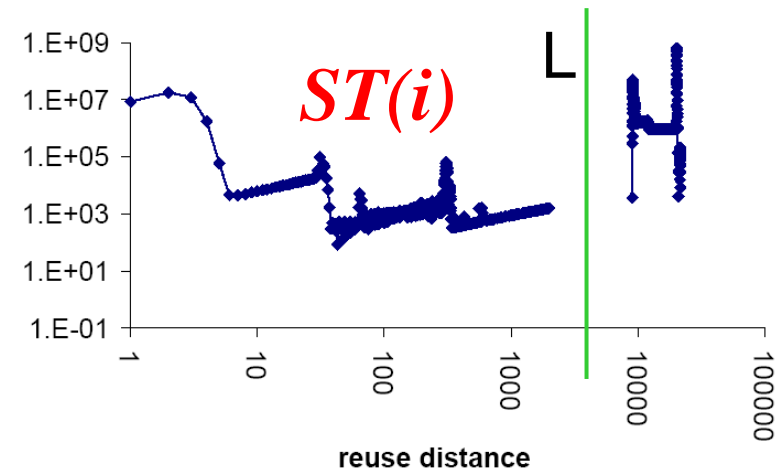
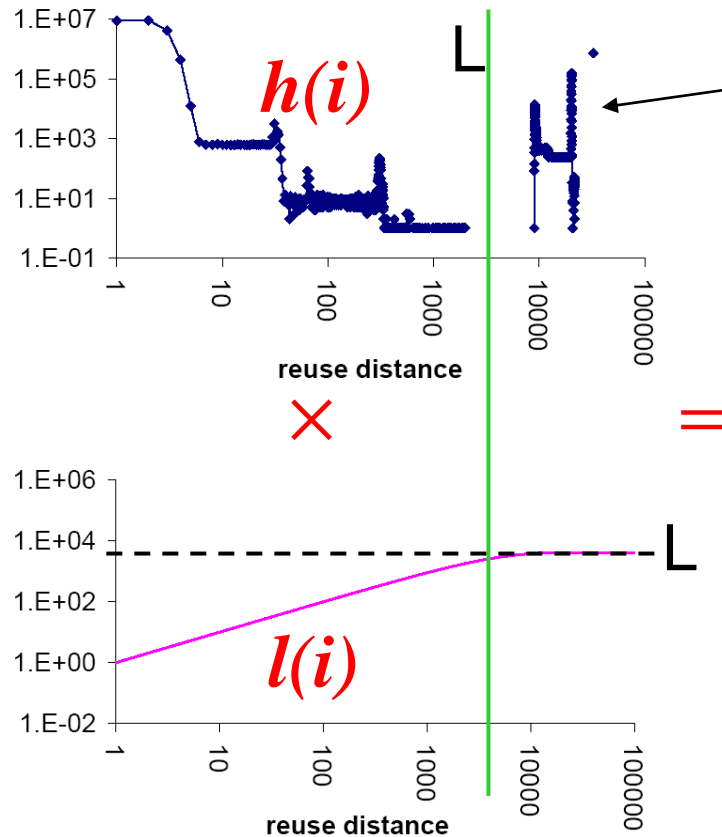


# Space time

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

# Spacetime

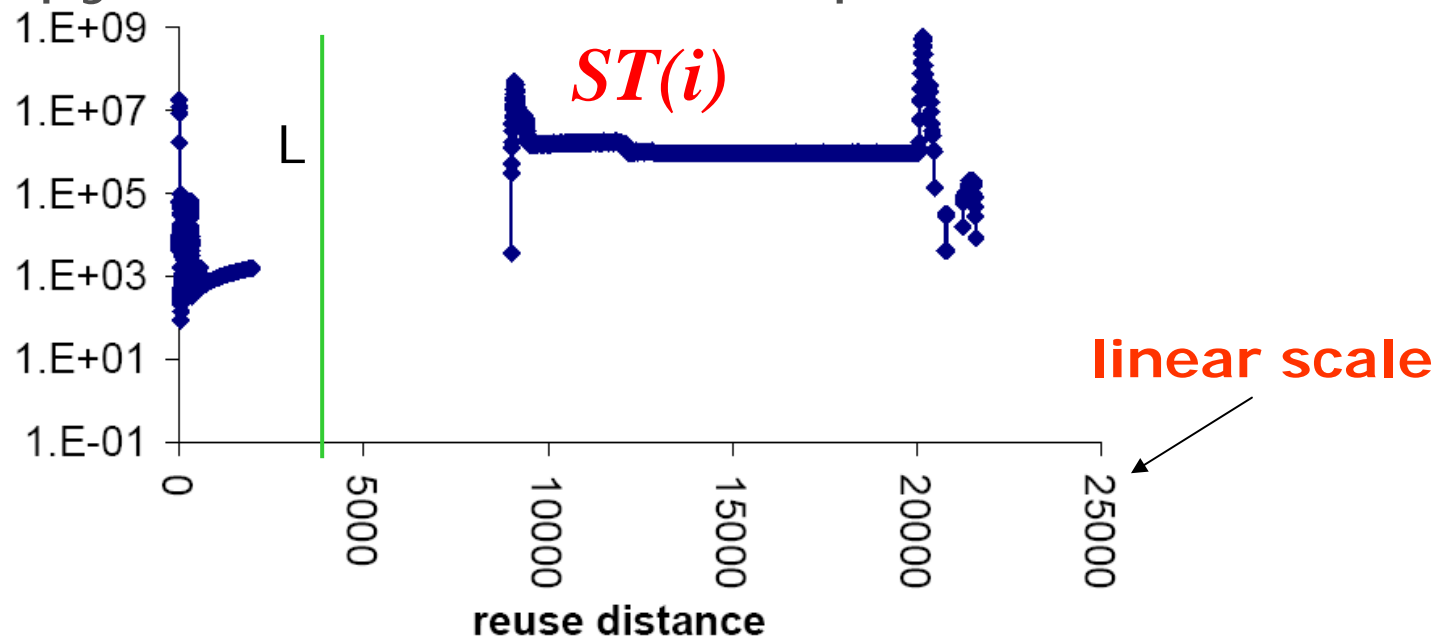
- Lifetime expectancy:  $l(i) = L \times f(i)$



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

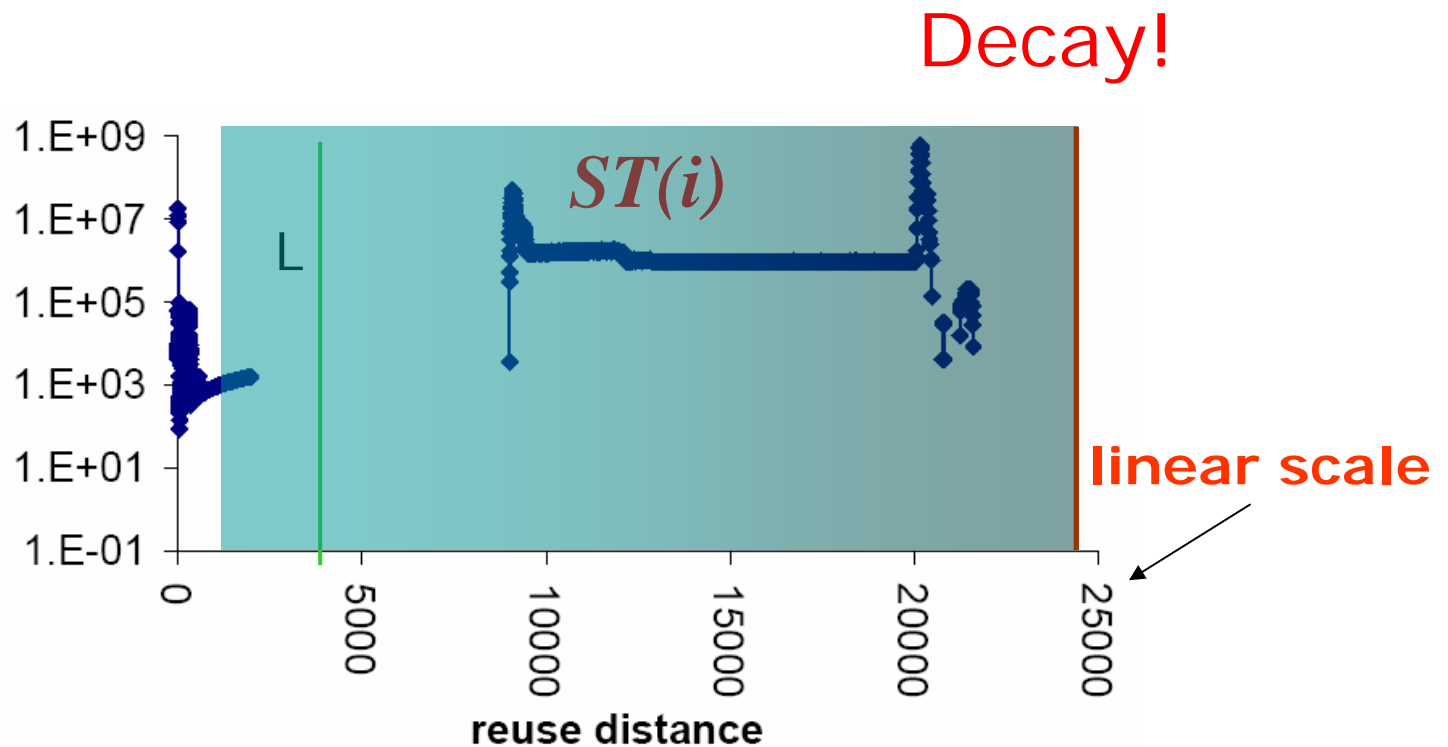
# Arguing for management

- 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**



---

# 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

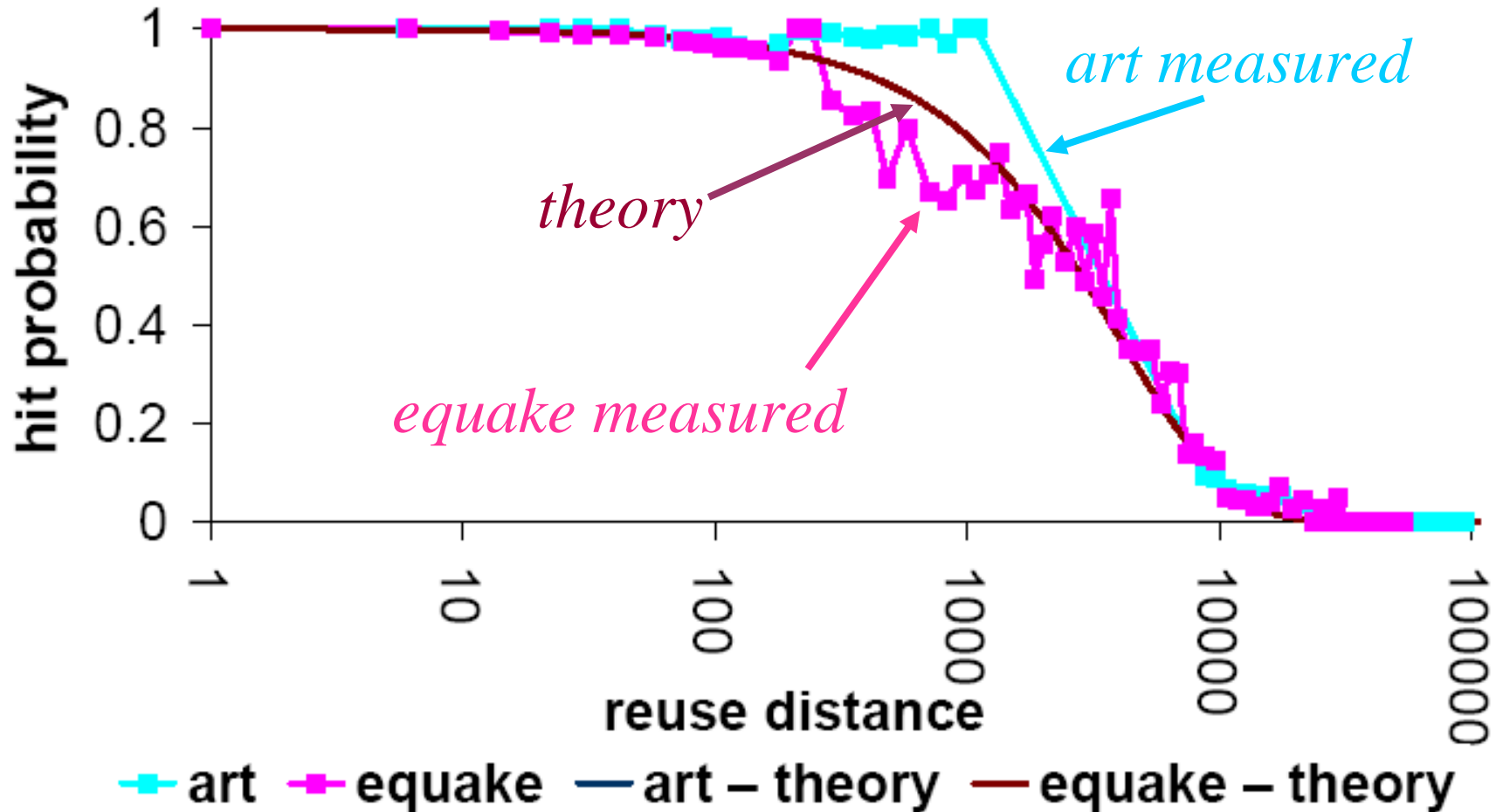
- Decay changes f-functions
  - Existence of decayed lines changes replacement policy
  - Not random anymore!
- Many different approximations possible ...
- **MURPHY** vs. managed replacements
  - Managed: finding a decayed line to replace
  - Murphy: using the underlying replacement policy (RR)
- Probability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

# 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  $L/\mu$**
  - Non-decayed applications “think” they share a much larger cache!
- For reuse-distances larger than the Decay Interval:
  - **$f = 1$  &  $f\text{-bar} = 0$**

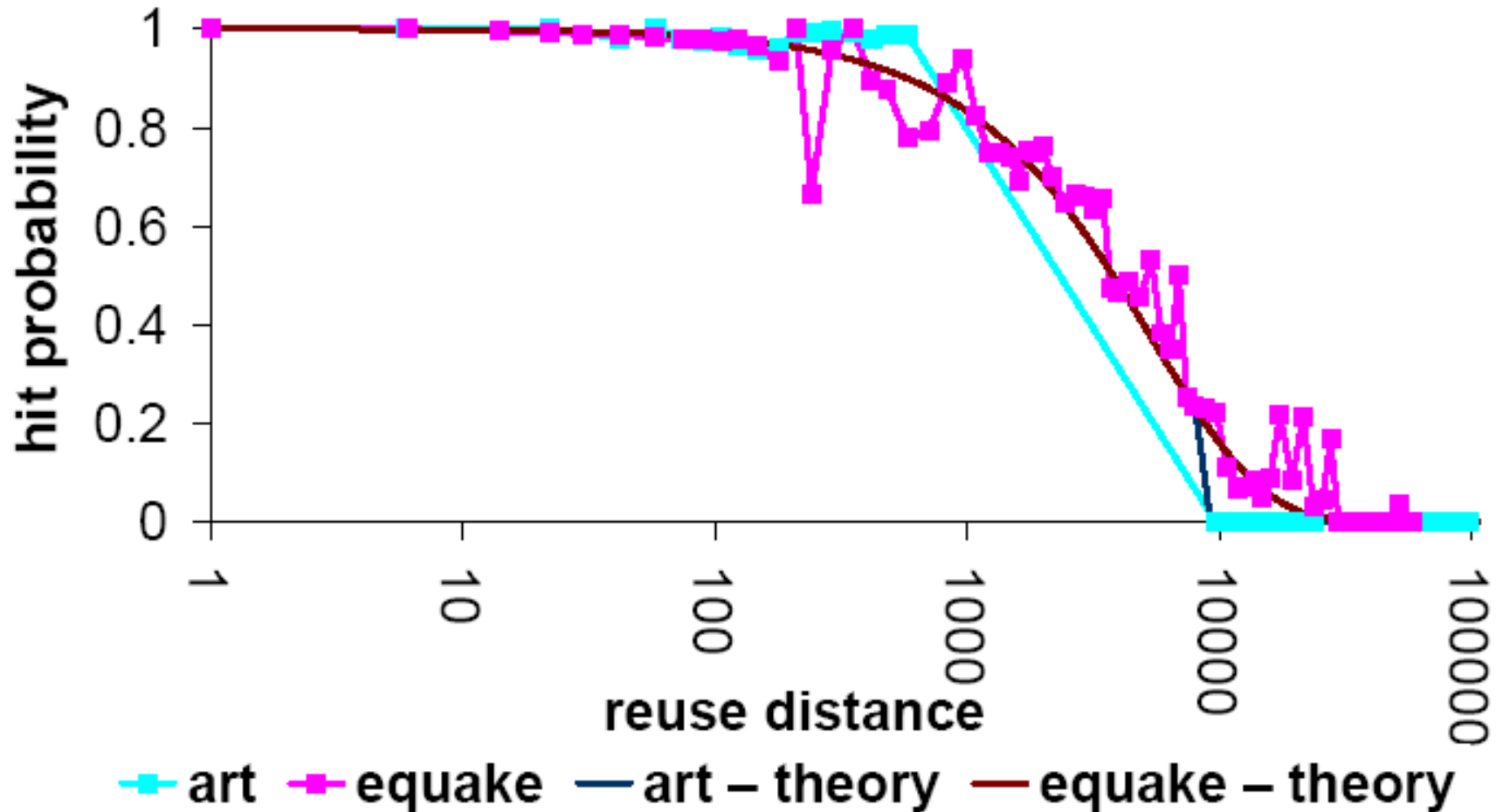
# Non-Decayed f-bar functions

f-bar: Size = 256KB – Decay Interval = Inf



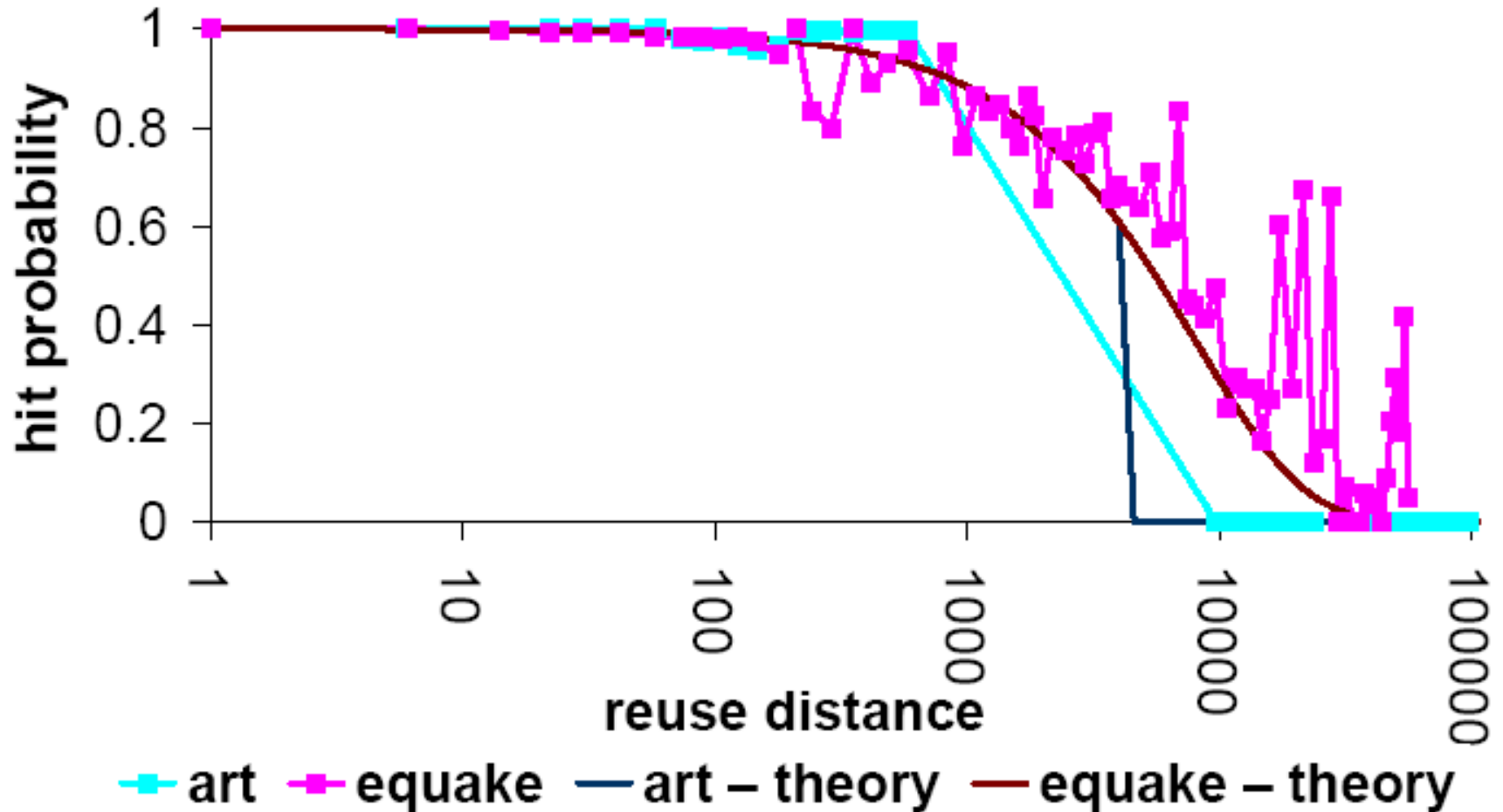
# Decayed f-bar functions

f-bar: Size = 256KB – Decay Interval = 8K CAT



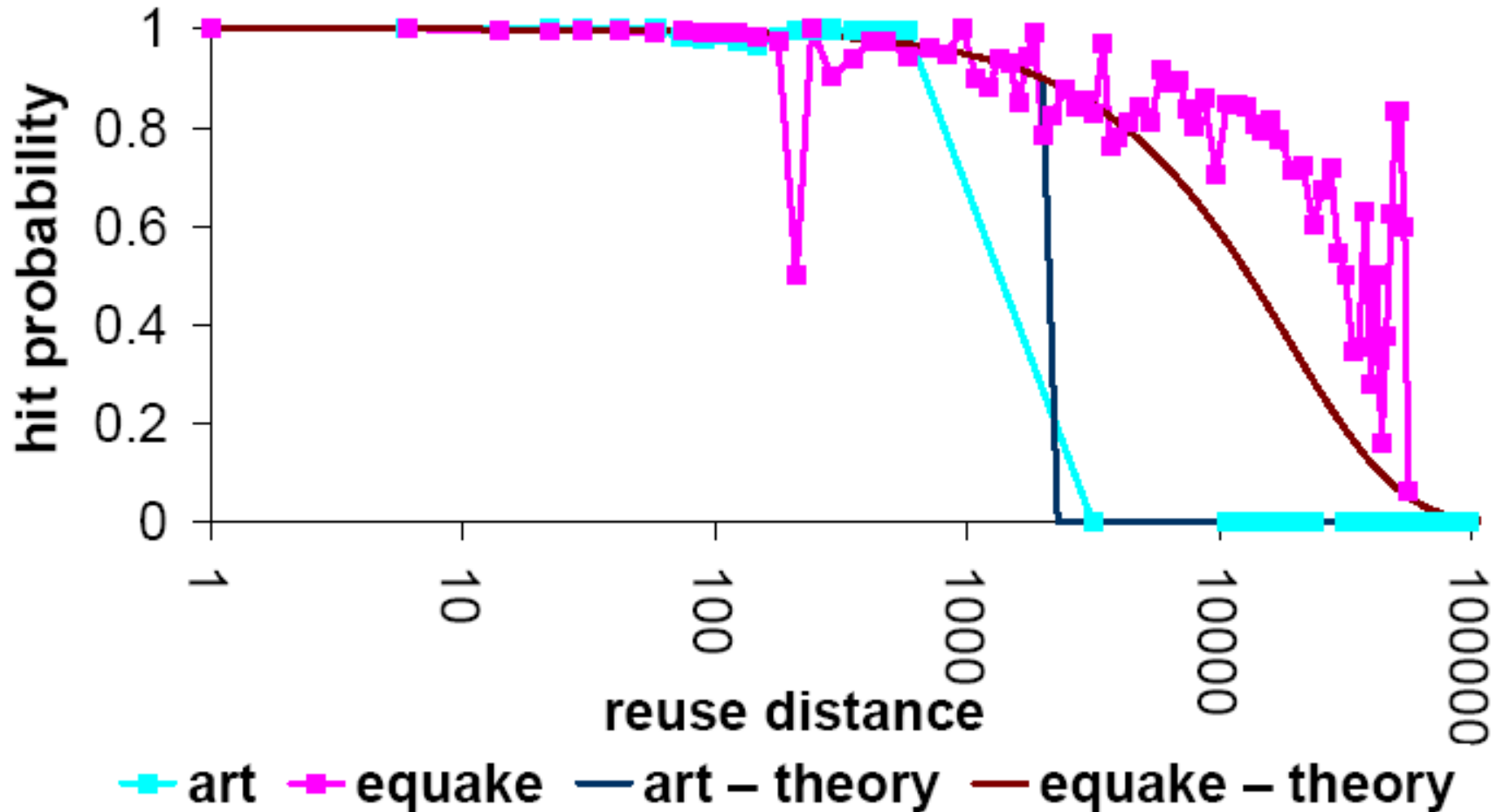
# Decayed f-bar functions

f-bar: Size = 256KB – Decay Interval = 4K CAT



# Decayed f-bar functions

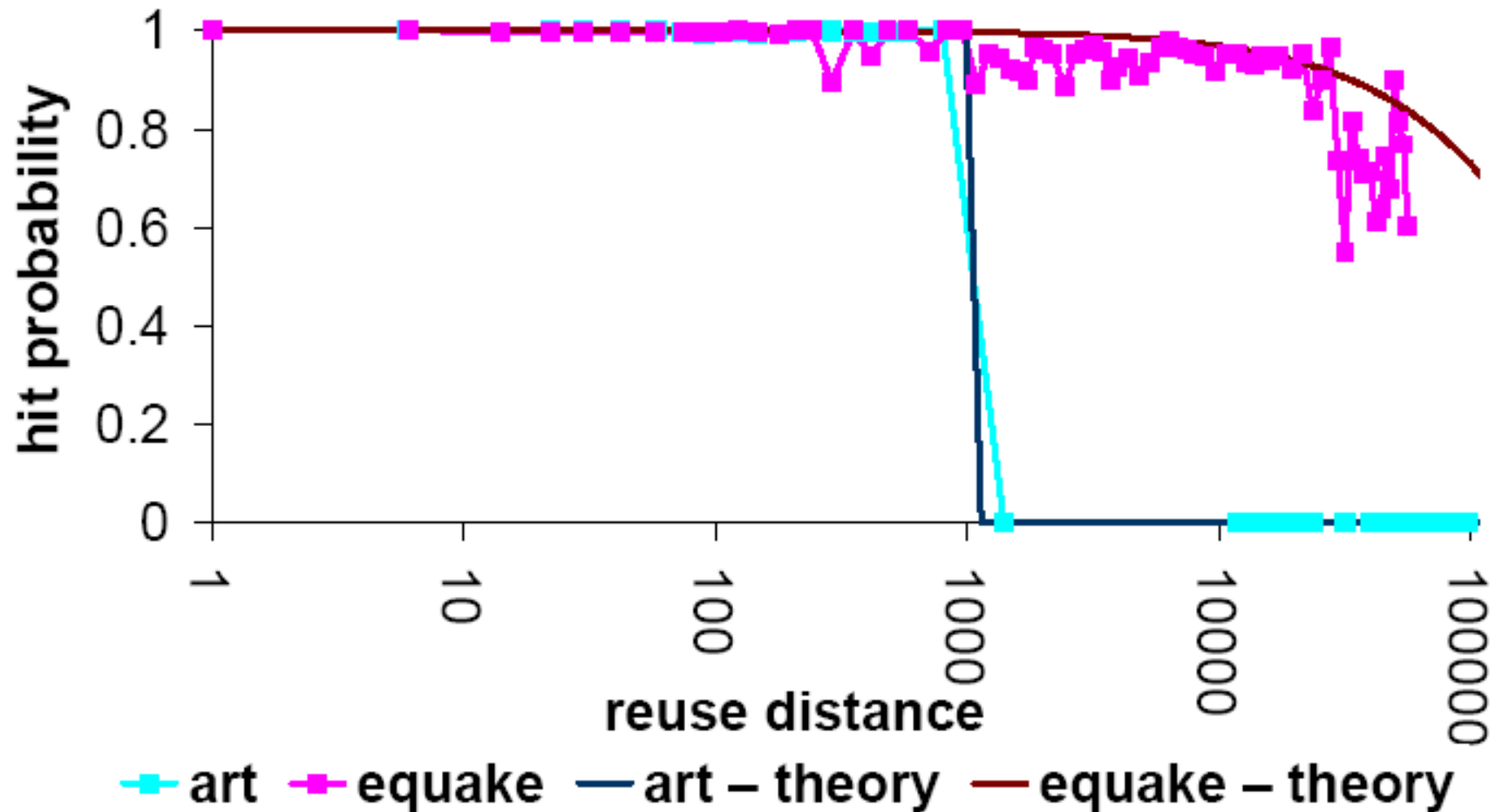
f-bar: Size = 256KB – Decay Interval = 2K CAT





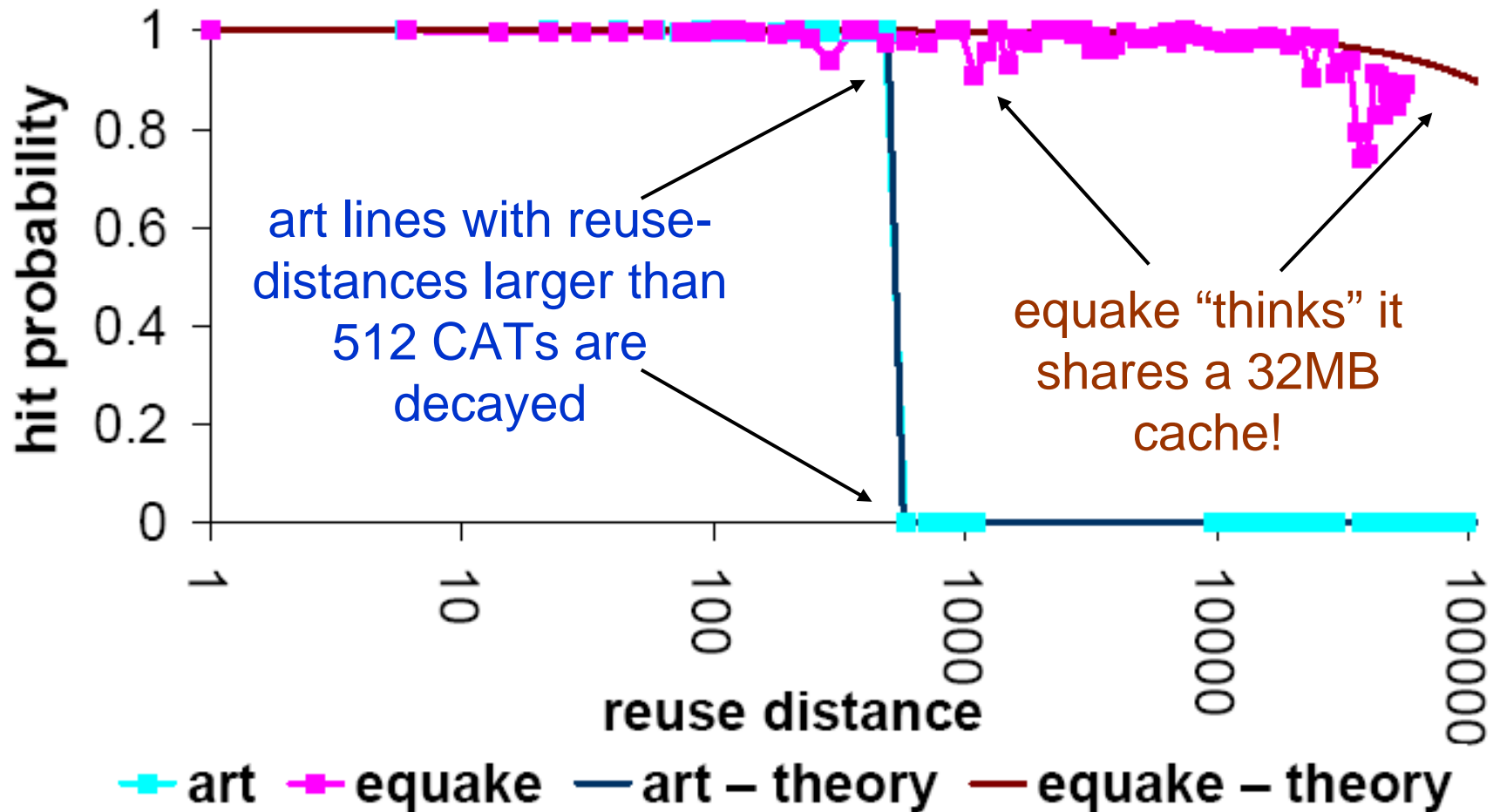
# Decayed f-bar functions

f-bar: Size = 256KB – Decay Interval = 1K CAT

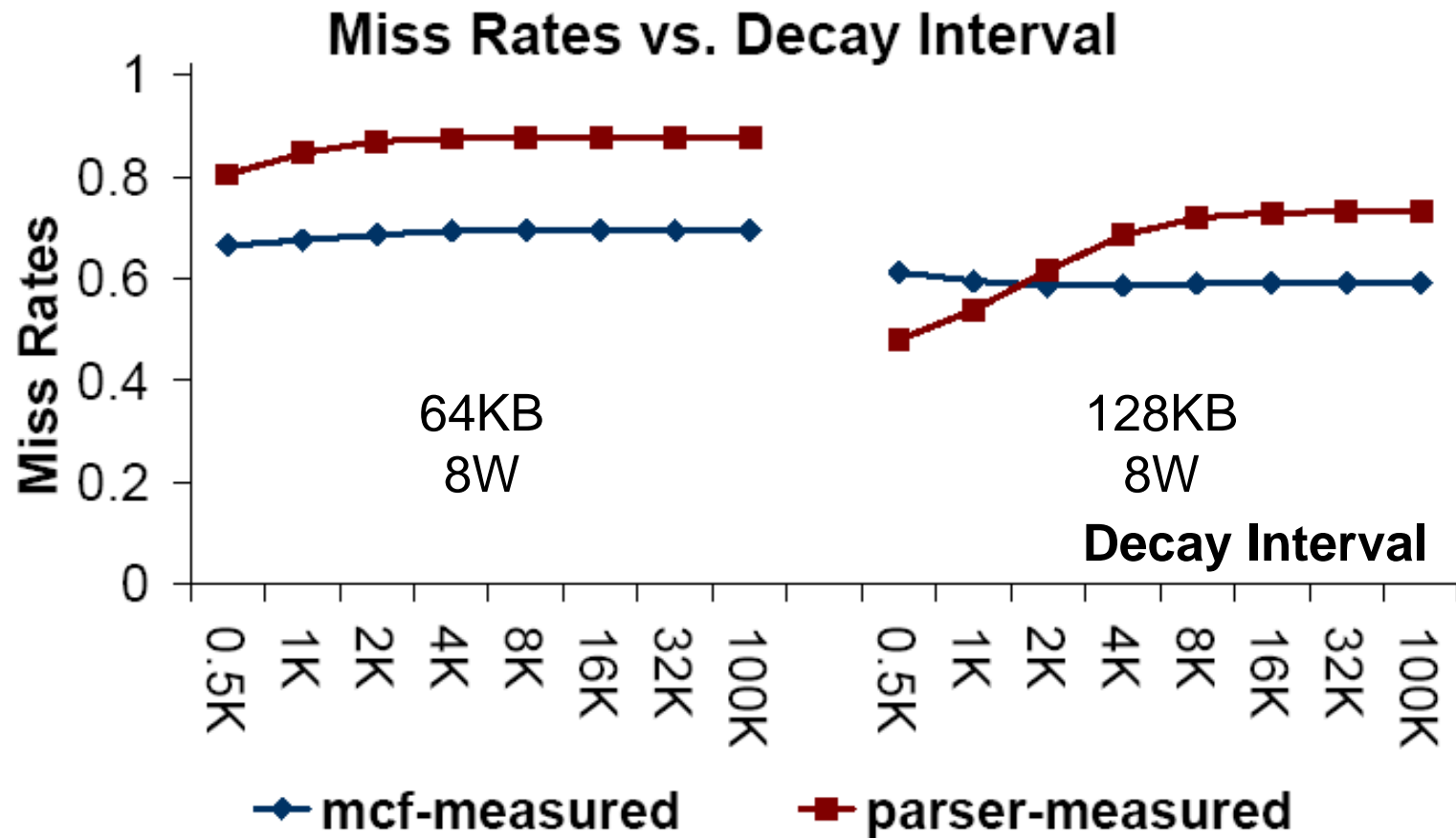


# Decayed f-bar functions

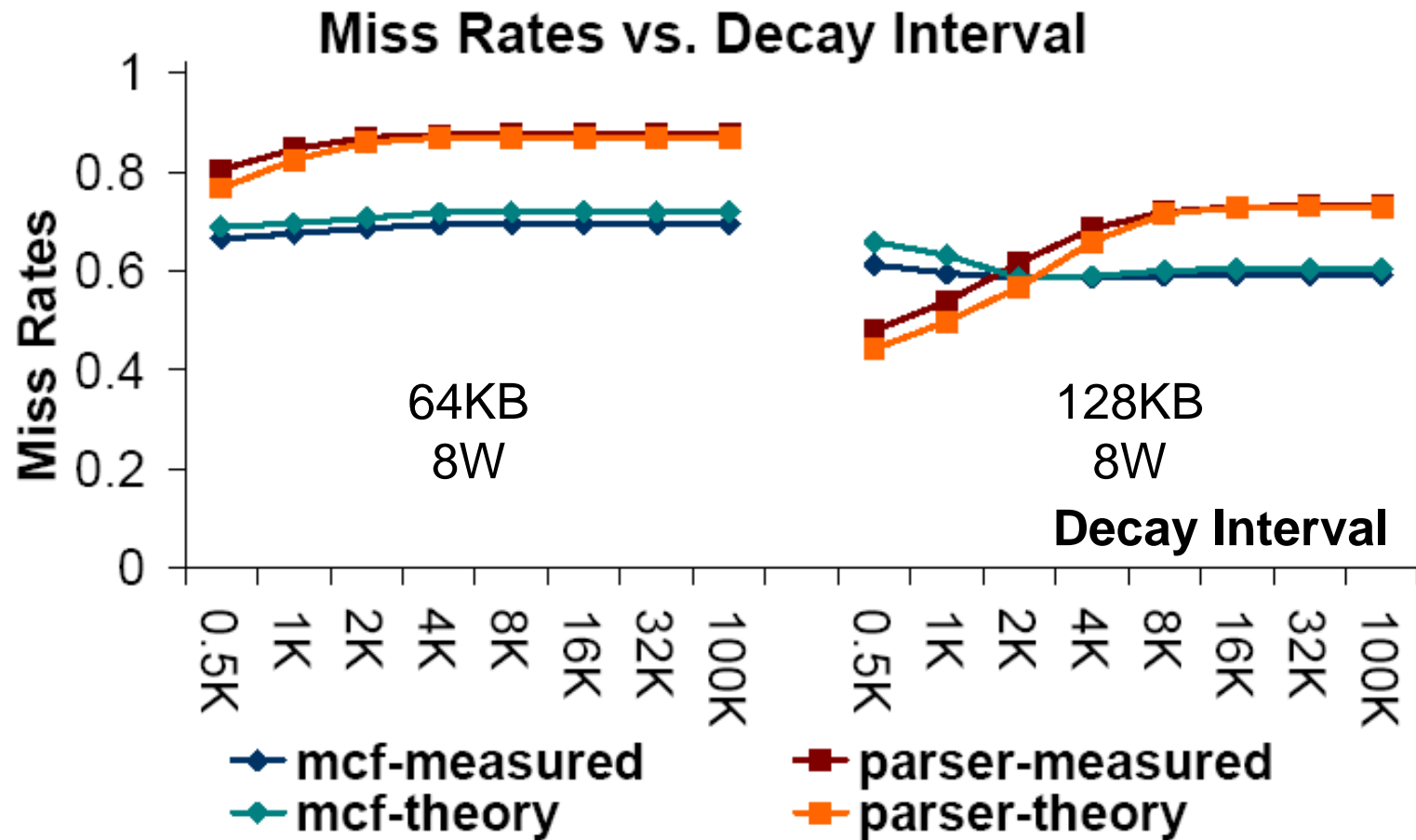
f-bar: Size = 256KB – Decay Interval = 512 CAT



# Model validation: mcf-parser – Miss Rates

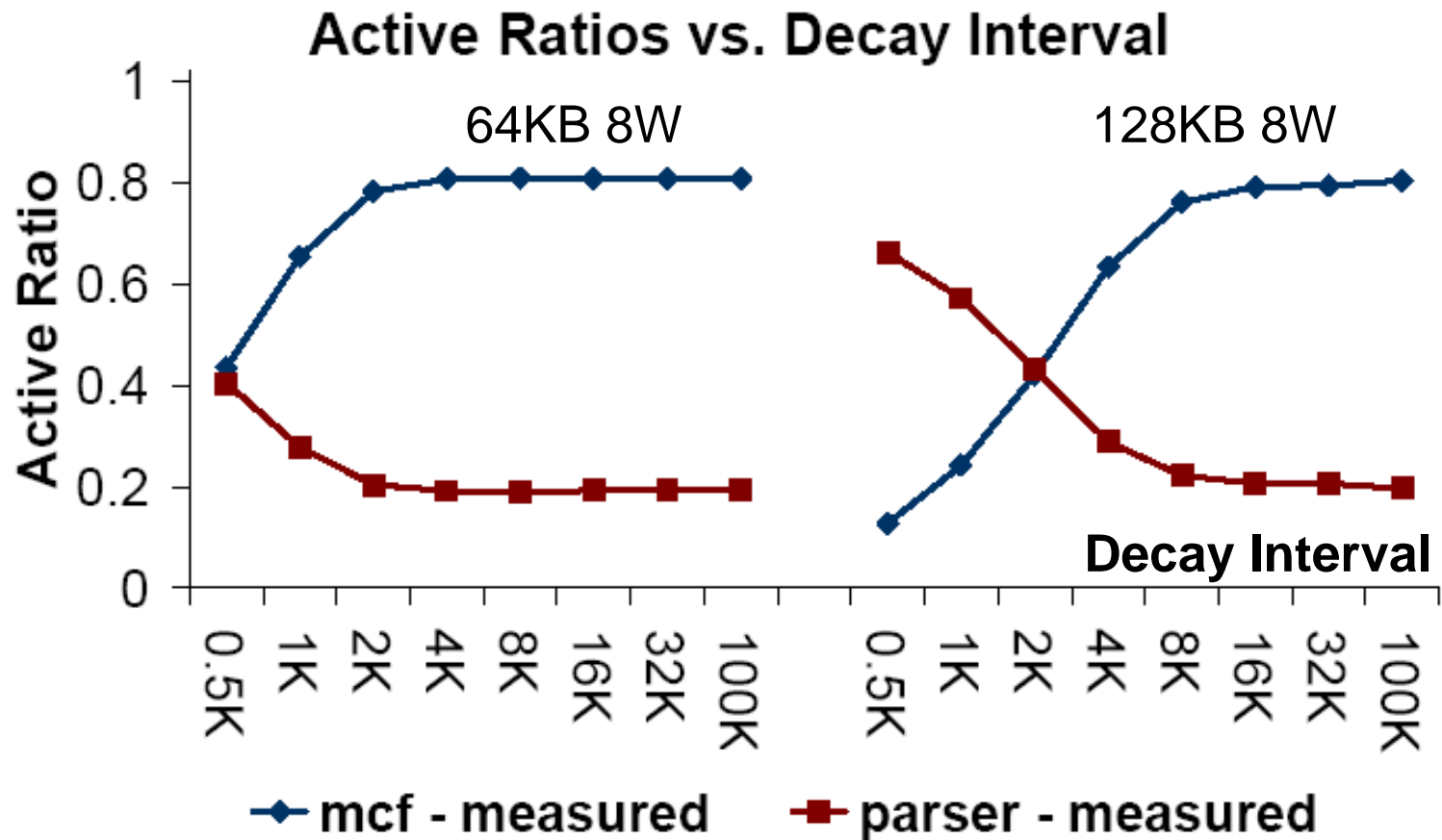


# Model validation: mcf-parser – Miss Rates

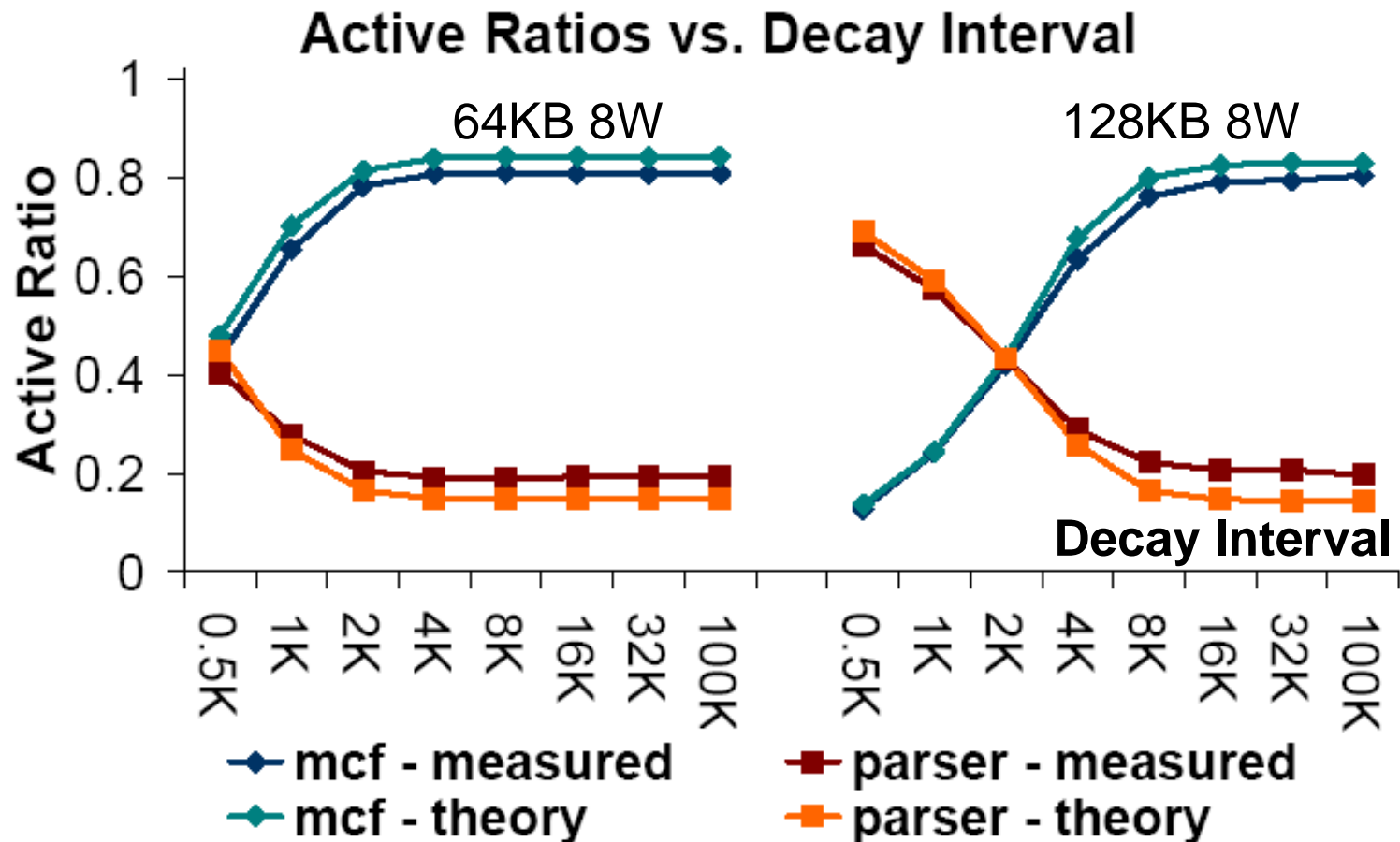


# Model validation:

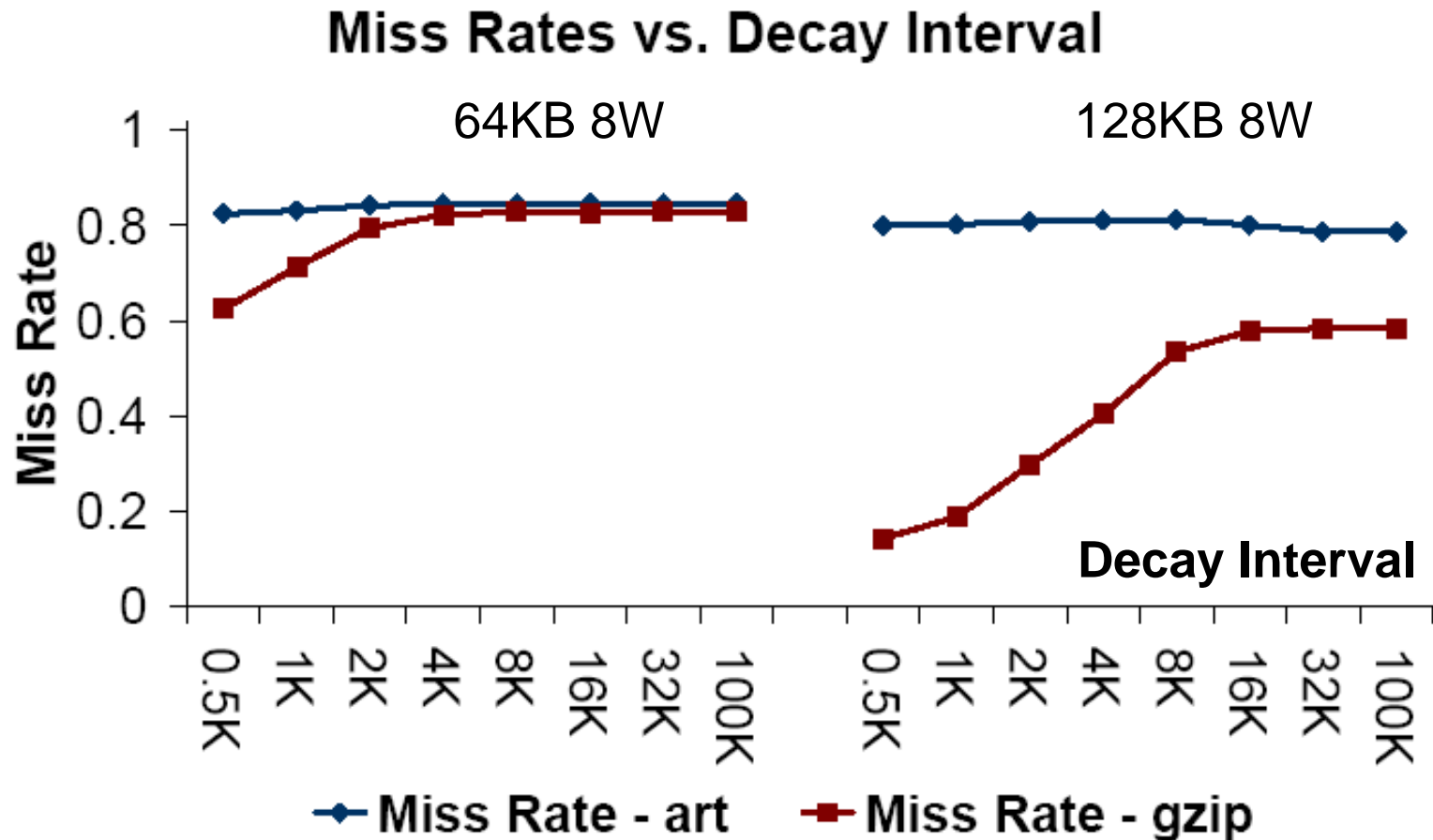
## mcf-parser – Active Ratios



# Model validation: mcf-parser – Active Ratios

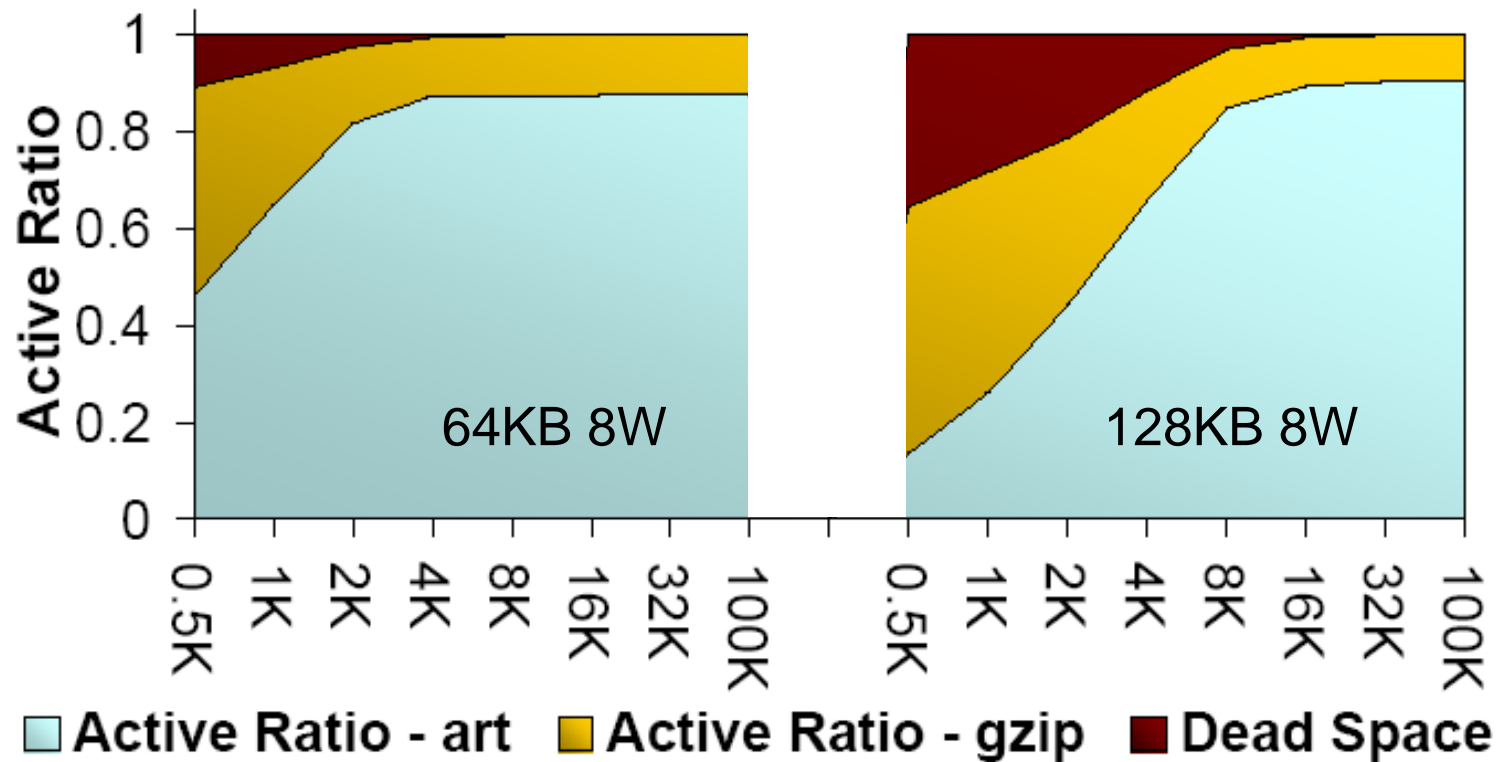


# Reasoning about cache management: art-gzip – Miss Rates



# Reasoning about cache management: art-gzip – Active Ratios

Active Ratios – Dead Space vs. Decay Interval





# 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

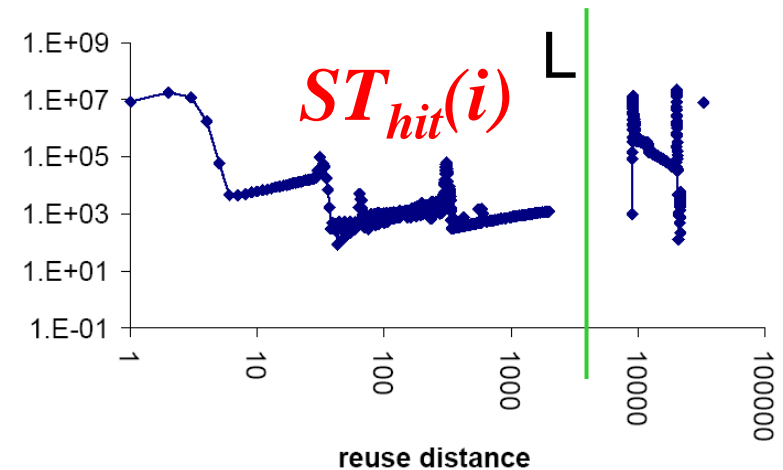
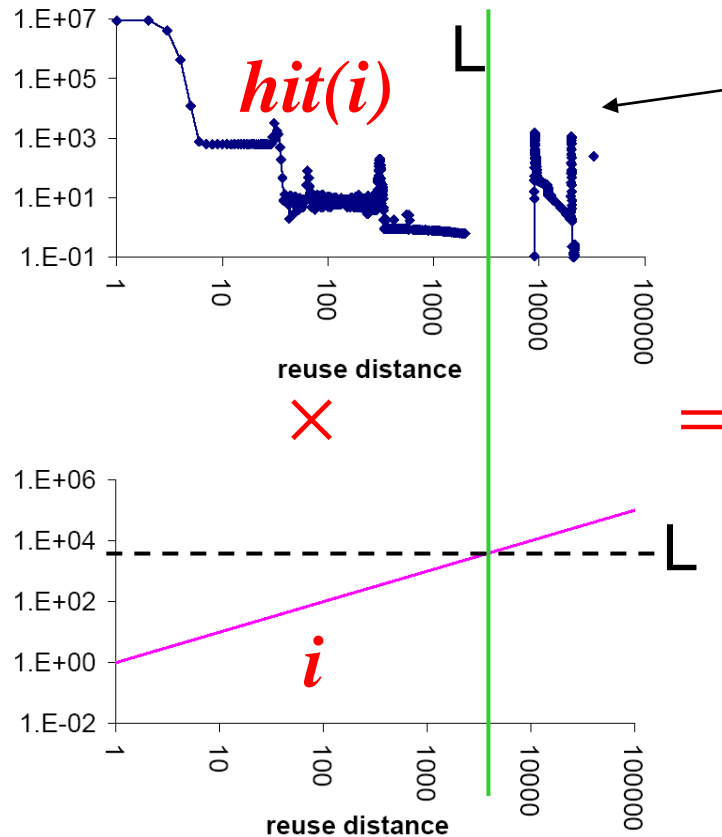
- Model:
    - Hits, misses, active ratios...
    - Can be computed @ runtime
  - Control Mechanisms
    - CAT Decay
    - Other cache partitioning approaches...
  - Use such info to implement high-level policies
-

# Questions



# Hits Spacetime

- Hits occupy ST for their reuse distance



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

# Misses Spacetime

- Misses occupy ST until their replacement

