Cache Replacement Based on Reuse-Distance Prediction

Georgios Keramidas, Pavlos Petoumenos, Stefanos Kaxiras

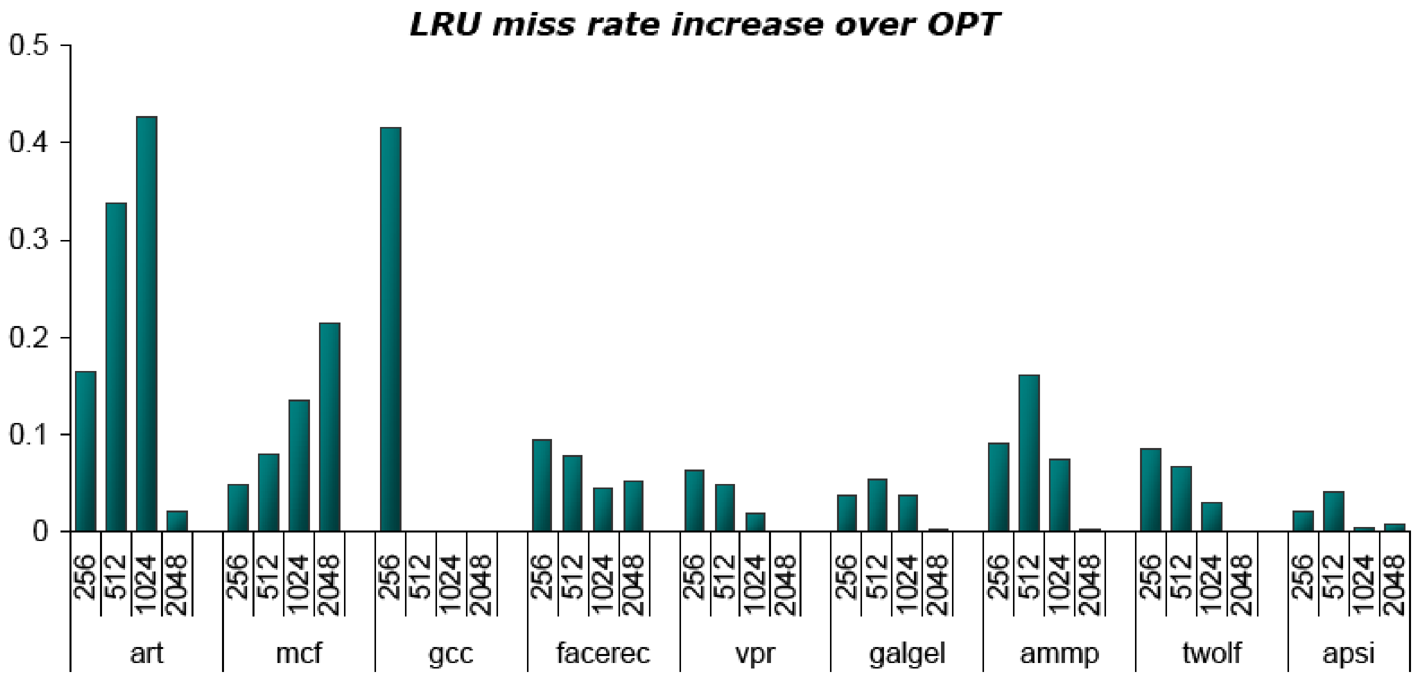
University of Patras, Greece

Introduction

- We need caches
  - The memory wall rises more and more

- Contributions:
  - Instruction based reuse distance prediction
  - Potential for cache level optimizations
  - Case study: a replacement algorithm for second level caches

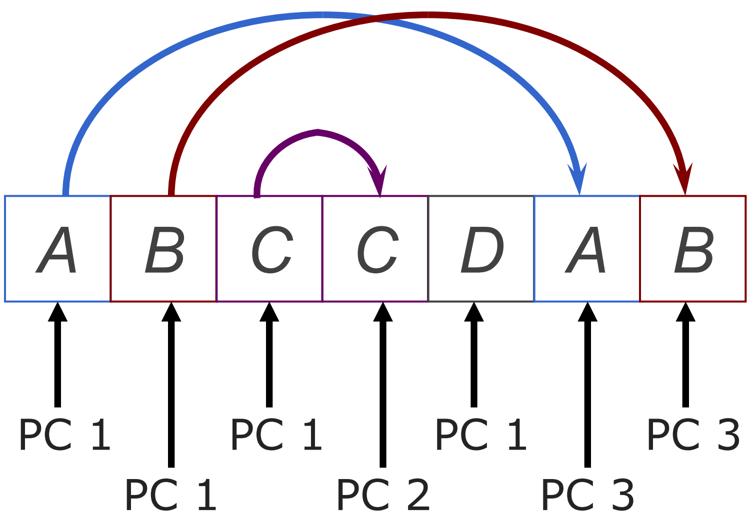
LRU vs Belady's Optimal Replacement



- LRU: Inefficient for L2 caches
- Reasons:
  - L1 filtering
  - Highly associative caches

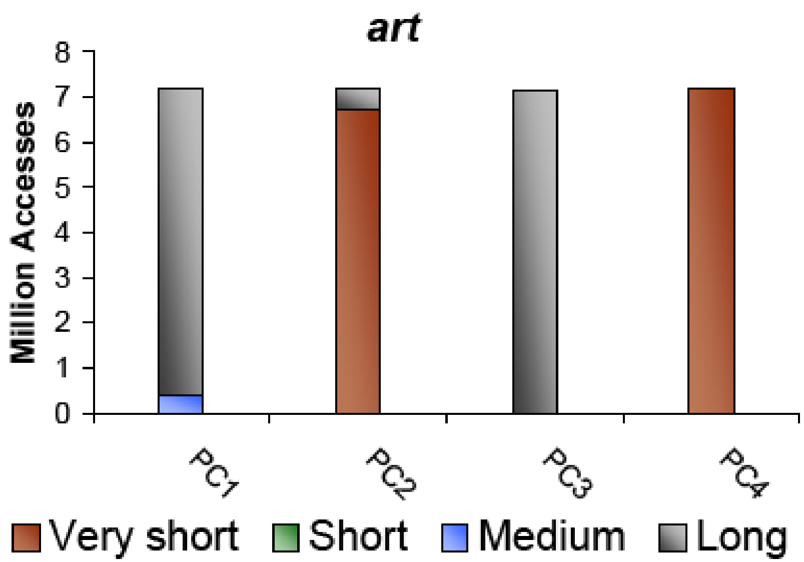
Can we see the future?

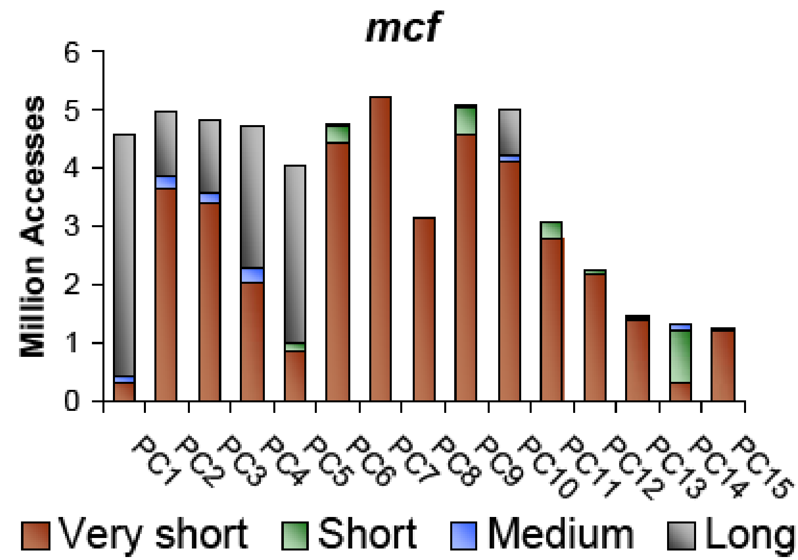
- Yes, via prediction
- Memory behavior → repeating patterns
- Our Motivation:
  - Strong correlation between instructions (PC) and moment of next access (reuse distance)



Reuse Distance of A = 4
Reuse Distance of B = 4
Reuse Distance of C = 0

Reuse distances are predictable

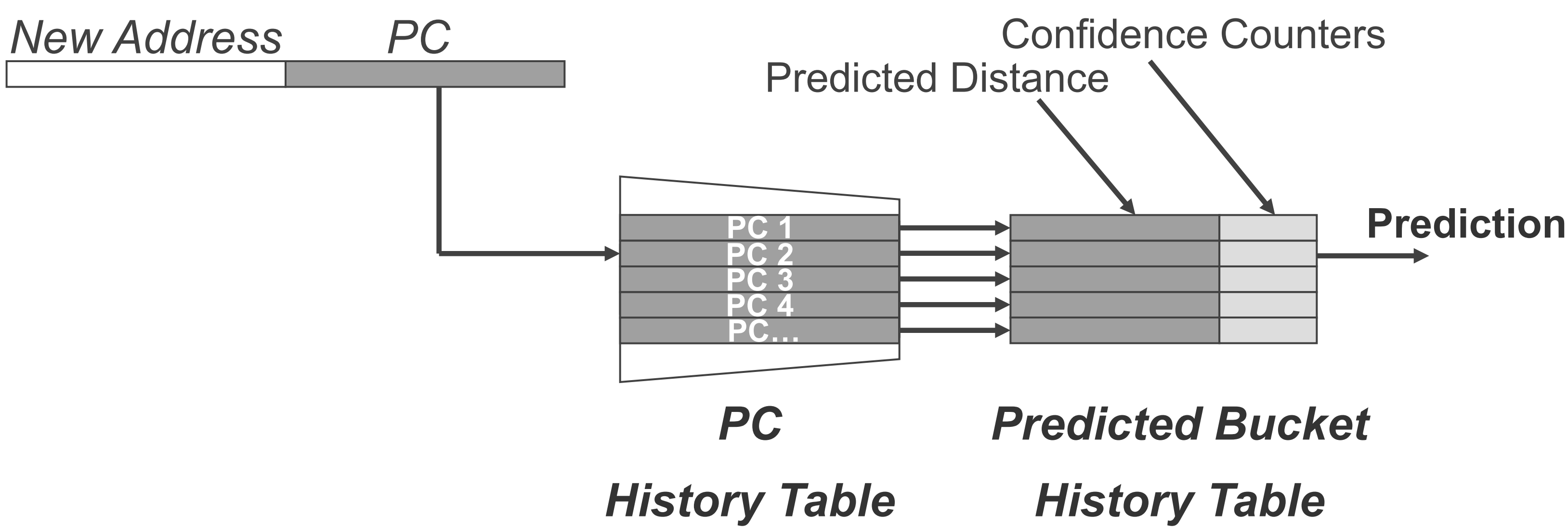




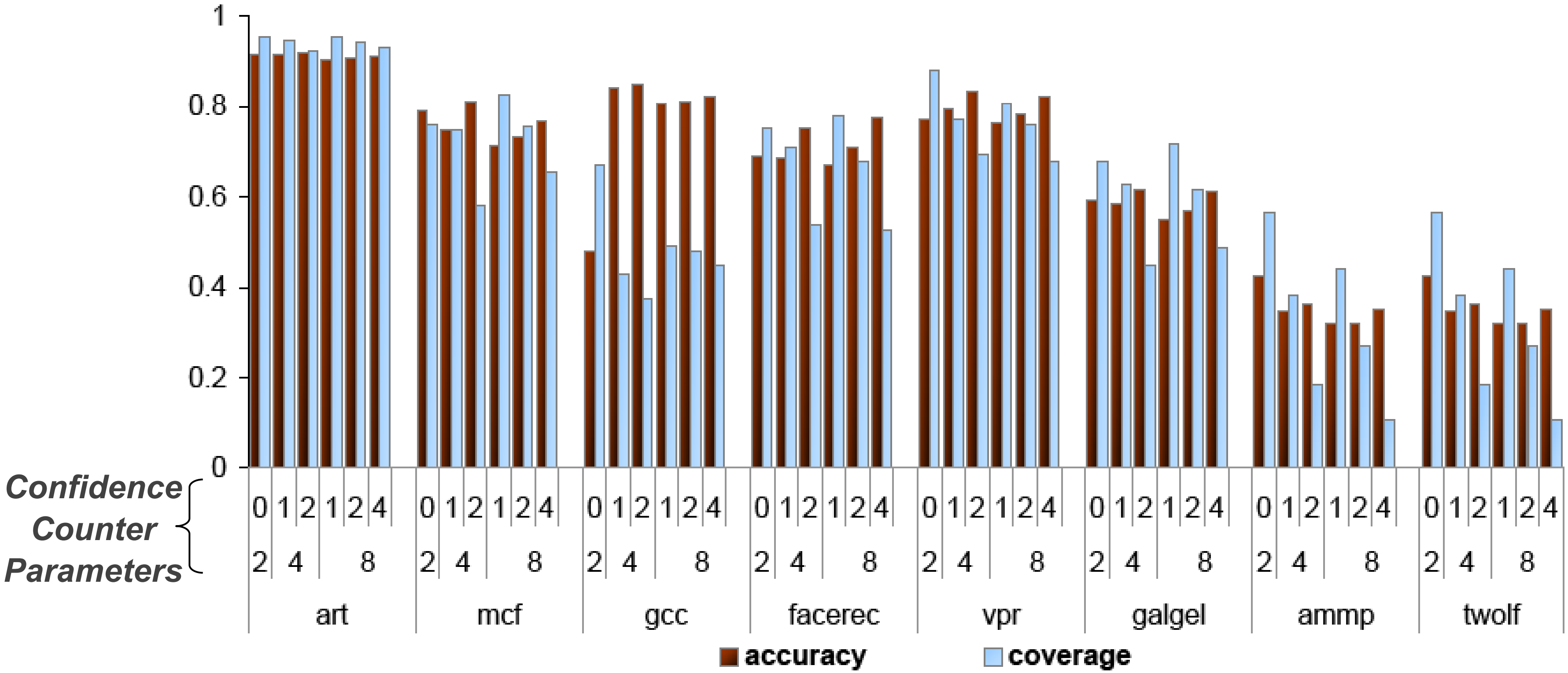
- Significant PCs:
  - Are few → Easily stored
  - Have predictable reuse distances

Instruction based Predictor

- Predicts the reuse distance of an access based on the PC which initiated the access
- Tracks the accessed line
- Upon reuse calculates the reuse distance
- Updates the predictor entry associated with the PC



Accuracy and Coverage

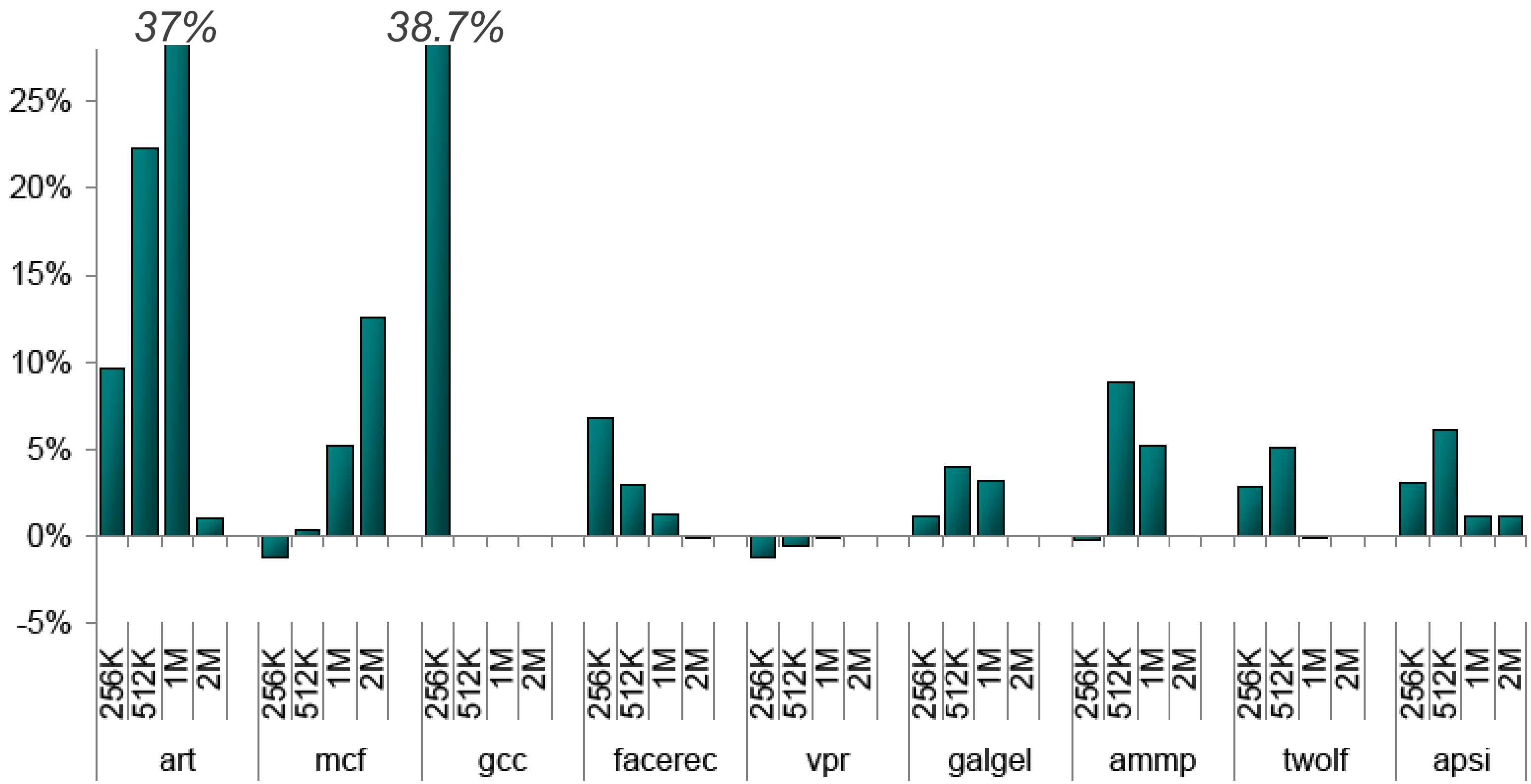


- Not bad, but also not perfect
  - The concept works
  - Even better implementations are possible

A case study: replacement policy

- Not all accesses are predictable
  - We cannot implement a perfect optimal algorithm
- Hybrid algorithm:
  - Replace the line used farthest in the future (OPT) when you have enough info
  - Replace the line used farthest in the past (LRU) if you don’t.
- More reliable predictor → policy closer to OPT

Miss rate reduction

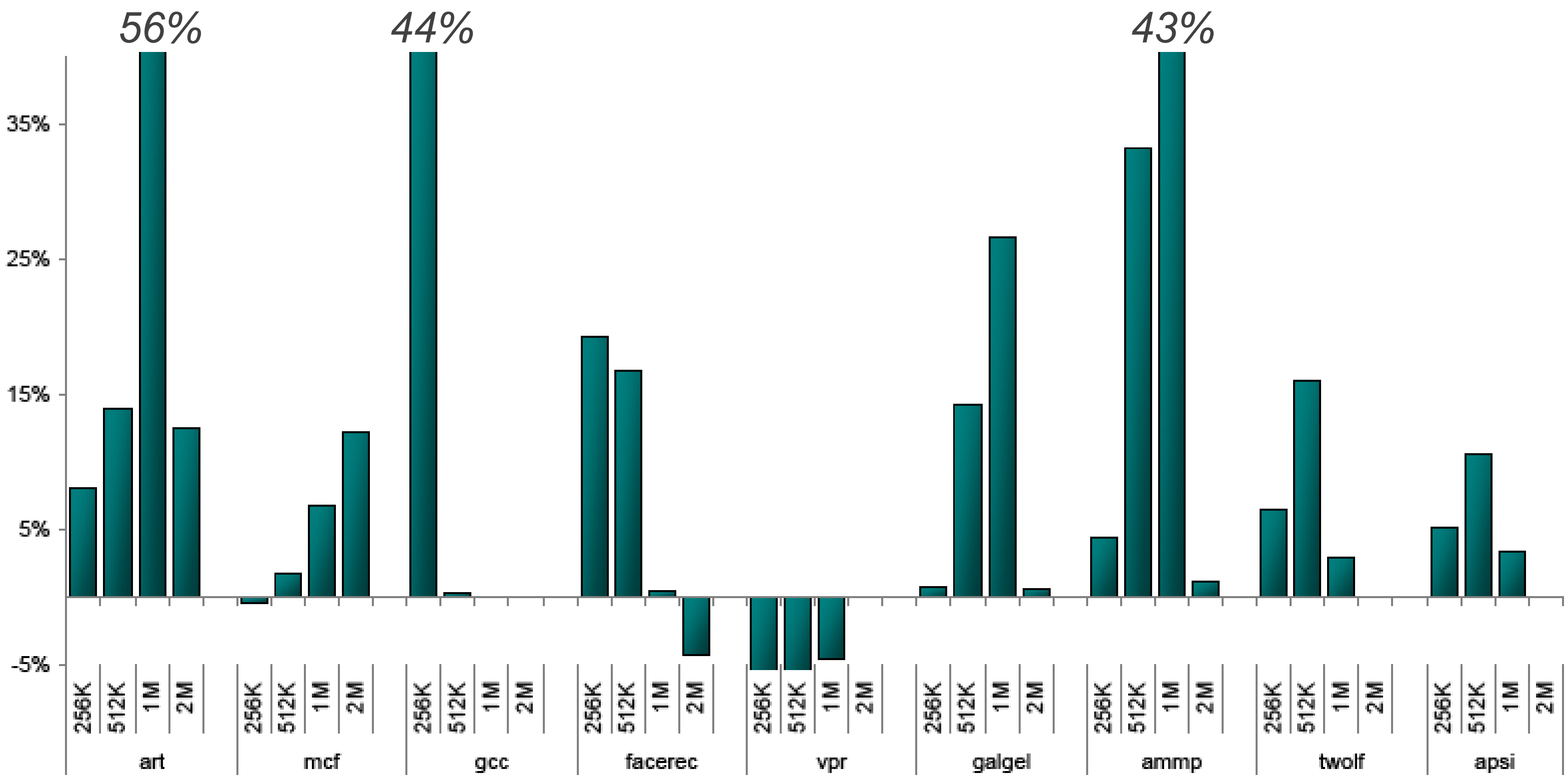


- Significant improvement
- Benchmarks with low accuracy are not greatly affected

Conclusions and Future Work

- Instruction based reuse distance predictor
  - Proof of concept
  - Low overhead (< 2%)
  - Good accuracy
- Replacement policy for L2 caches
  - Speed ups in many benchmarks
  - 60% in art
  - 43% in ammp
  - Slowdown in vpr
- Future Work:
  - Improve the effectiveness of the predictor
  - More cache level optimizations

Speedup (IPC)


56%
44%
43%
35%
25%
15%
5%
-5%
256K
512K
1M
2M
art
mcf
gcc
facerec
vpr
galgel
ammp
twolf
apsi

Dead Block Prediction vs Reuse Distance Prediction

Reuse Distance Prediction

- Predicts when the line is going to be accessed again
- Fine grained prediction
- Can identify which line is less needed
  - Very large RDs indicate dead lines
- More information can be extracted
- Proof of concept predictor

Dead Block Prediction

- Predicts when the line is going to be evicted
- Coarse grained prediction
- Can identify only dead lines
  - Not which is less needed
- Less information is stored
- Sophisticated predictor