On the Theory and Potential of LRU-MRU Collaborative Cache Management

Xiaoming Gu and Chen Ding

06.04.2011

UNIVERSITY *of* ROCHESTER

Outline

- Introduction
- Theoretical properties of LRU-MRU cache
- Potential of LRU-MRU collaborative caching
- Summary

UNIVERSITY of ROCHESTER

Motivation

- Cache management is important
  - the disparity between CPU and main memory
  - an off-chip memory access (aka. cache miss) is very slow
  - use on-chip cache to overcome

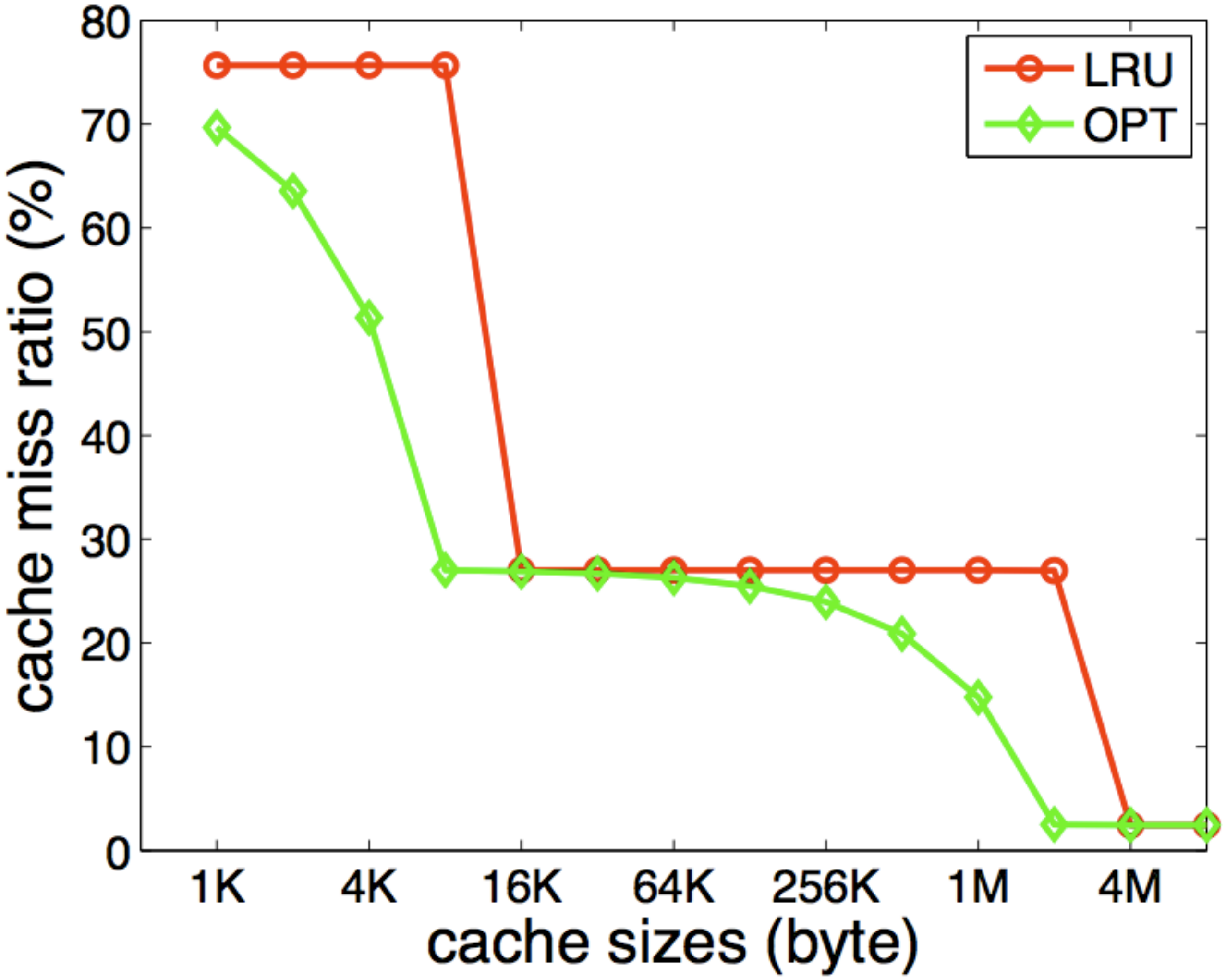
How to use cache more effectively?

Cache Replacement Algorithm

- A cache replacement algorithm decides which data are evicted
- LRU
  - victim is the one at LRU position
  - deployed in real cache but not optimal
- OPT
  - victim is the one that will be reused in the farthest future
  - optimal but not practical

UNIVERSITY of ROCHESTER

The Gap between LRU and OPT



SOR from SciMark 2.0

- Gradual change for the OPT miss ratio curve
  - abrupt for LRU
- Non-uniform gap
  - can be very large

BRIDGE THE GAP

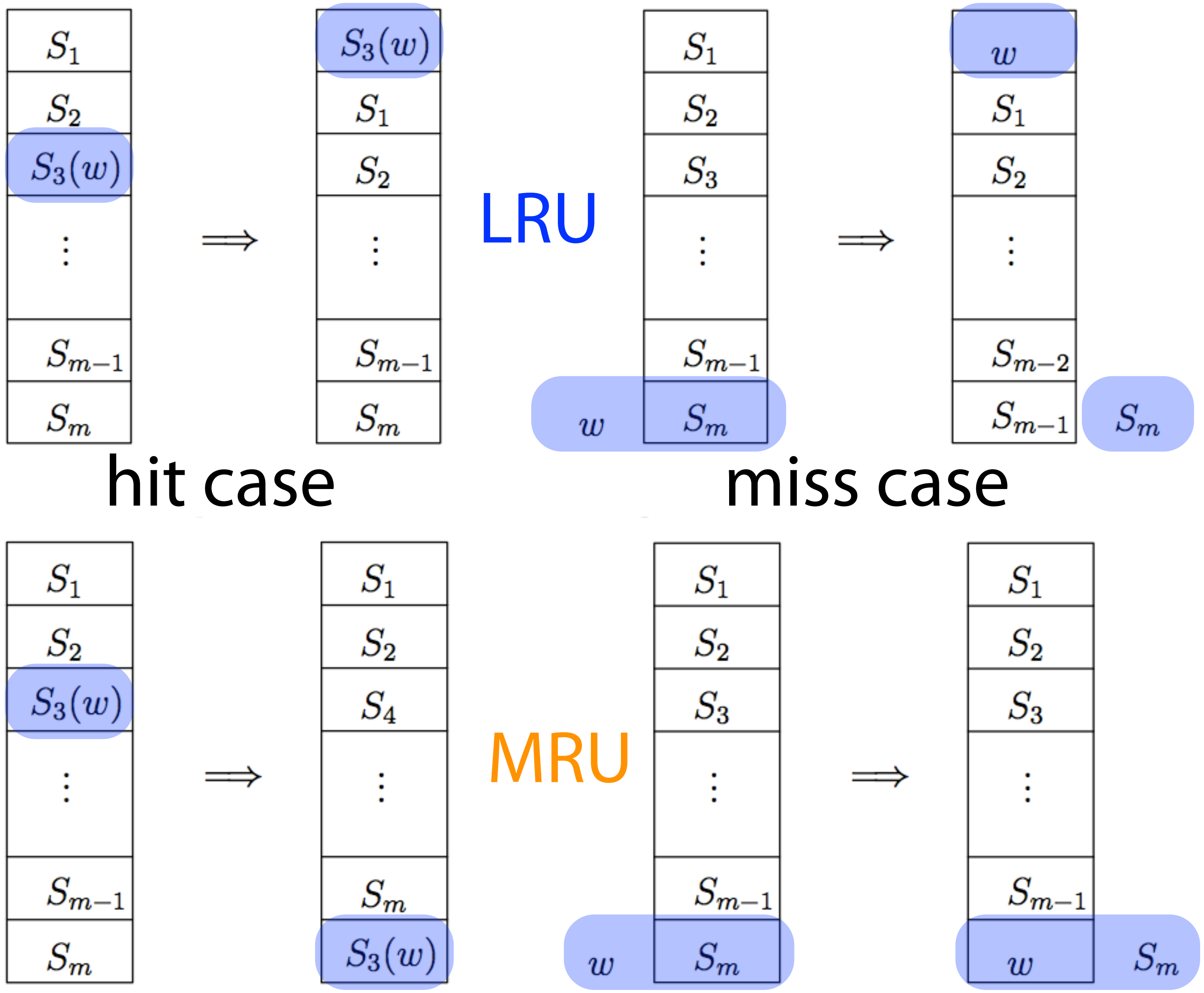
UNIVERSITY of ROCHESTER

Collaborative Caching

- Collaborative caching
  - the term was coined by Wang et al. in 2002
  - hardware provides multiple caching methods
  - software decides the right caching method for every access

LRU-MRU Collaborative Caching

- Two caching methods: LRU & MRU



Outline

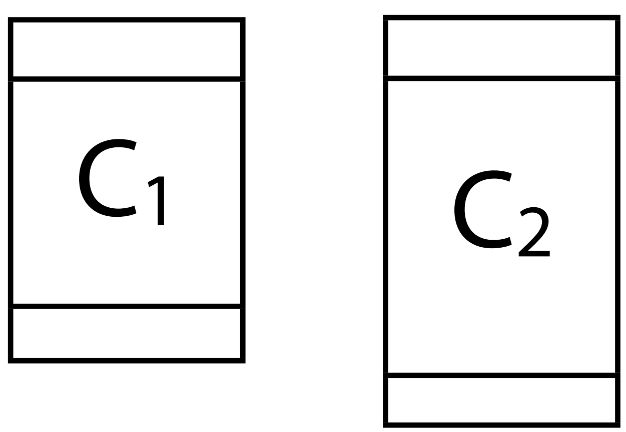
- Introduction
- Theoretical properties of LRU-MRU cache
- Potential of LRU-MRU collaborative caching
- Summary

Inclusion Property

- Inclusion property: the content of a smaller cache is always contained in a larger cache [Mattson et al., 1970]
  - cache miss ratio keeps non-increasing with larger cache sizes
  - LRU & OPT both have inclusion property

LRU-MRU Cache Has Inclusion Property

- Inductively proved that inclusion property is satisfied



$|C_1| < |C_2|$

$content(C_1) \subseteq content(C_2)$ after every access a_i

  - the base step
  - the inductive step

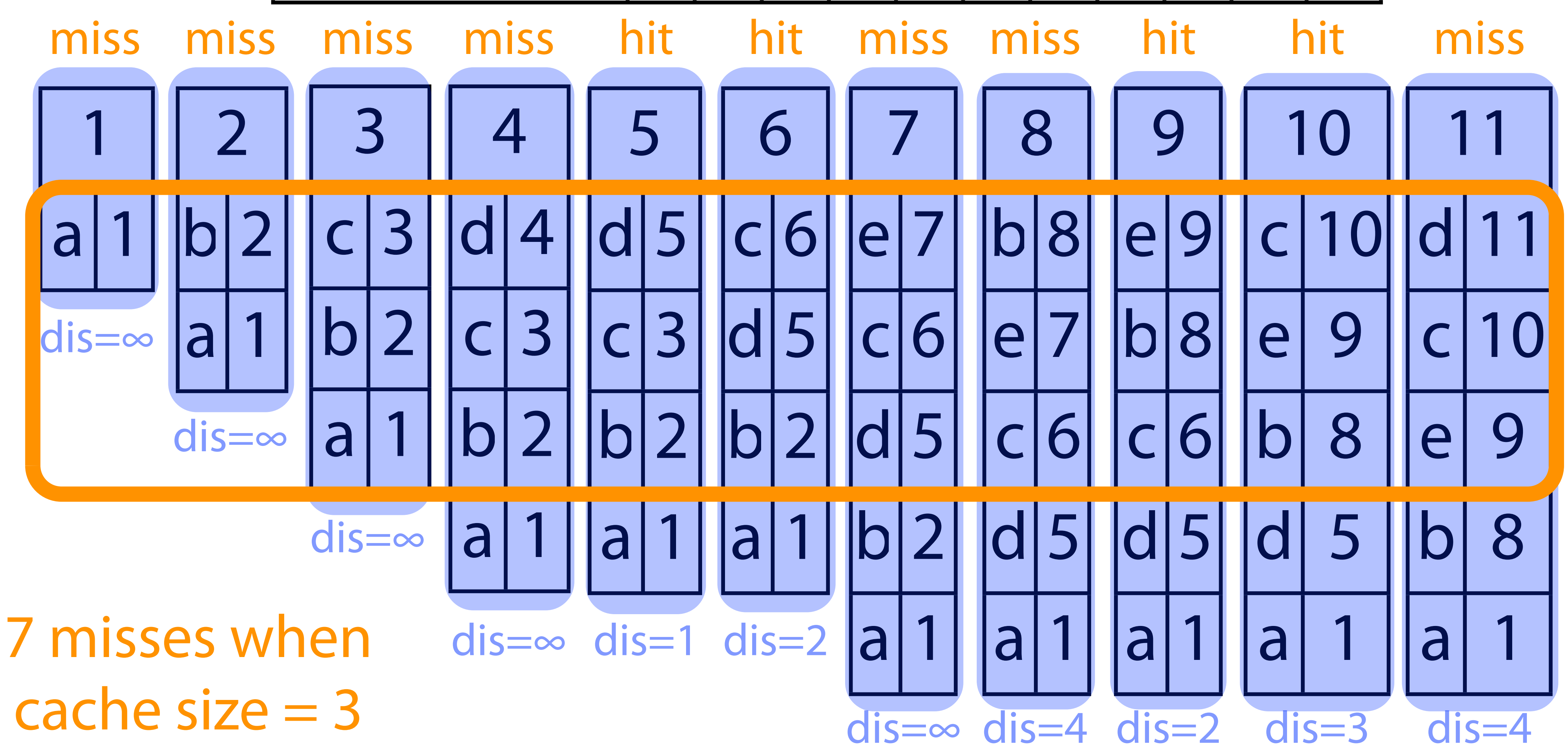
Stack Distance

- Stack distance is the minimal cache size to make an access become a hit
  - inclusion property is the precondition
- One-pass stack distance analyzer
  - simulates all cache sizes at the same time
  - the core is to maintain a priority list
    - LRU: the priority is the current access time
    - OPT: the priority is the next access time

UNIVERSITY of ROCHESTER

An Example of LRU Stack Distance Computation

access No.	1	2	3	4	5	6	7	8	9	10	11
data	a	b	c	d	d	c	e	b	e	c	d



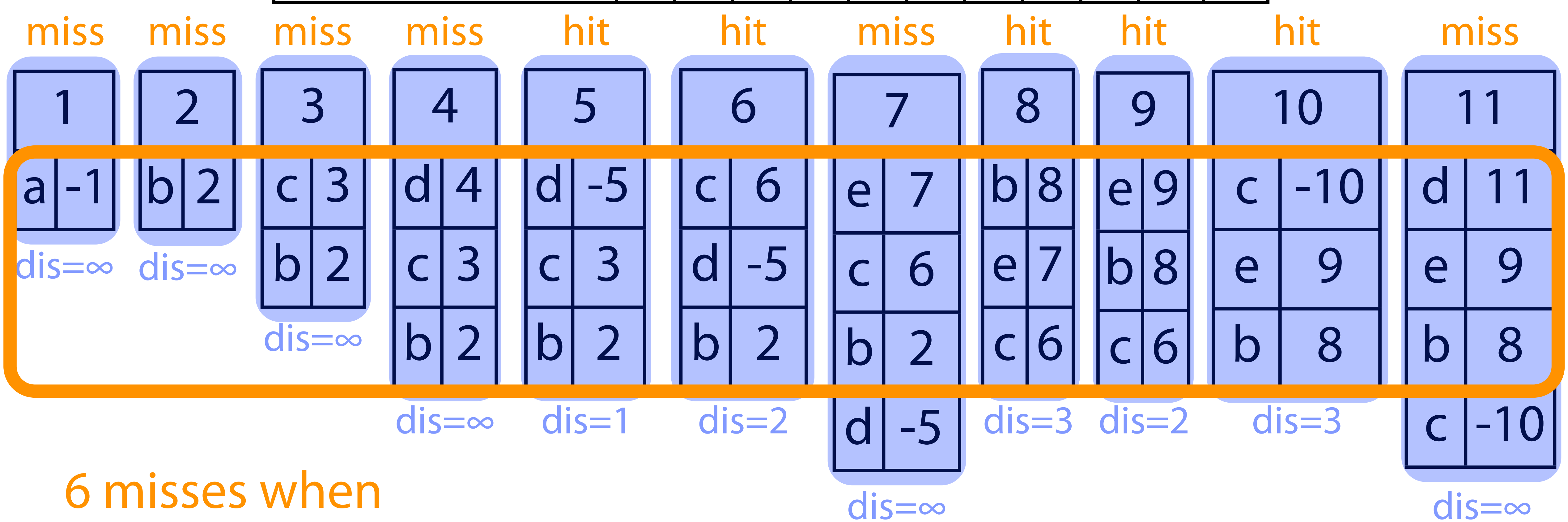
The Algorithm for LRU-MRU Stack Distance

- Based on the general one [Mattson et al., 1970]
- The most significant change---the priority is a variant of access time
  - the current access time for LRU
  - the negation of the current access time for MRU

UNIVERSITY of ROCHESTER

An Example of LRU-MRU Stack Distance Computation

access No.	1	2	3	4	5	6	7	8	9	10	11
data	a	b	c	d	d	c	e	b	e	c	d
LRU or MRU?	M	L	L	L	M	L	L	L	L	M	L



6 misses when cache size = 3

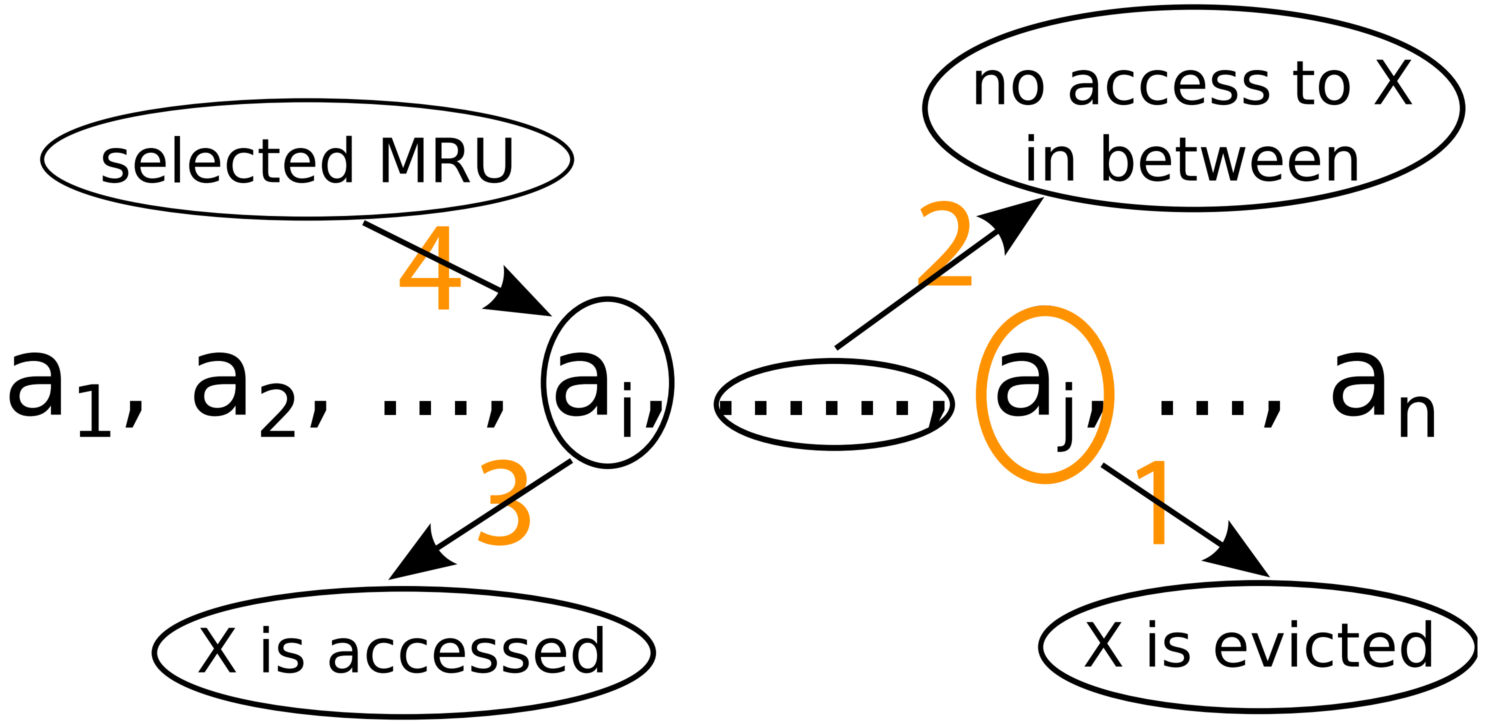
UNIVERSITY of ROCHESTER

Outline

- Introduction
- Theoretic properties of LRU-MRU cache
- Potential of LRU-MRU collaborative caching
- Summary

LRU-MRU Cache Can Be Optimal

- Do MRU selection based on an OPT simulation
  - at the beginning, all accesses use LRU
  - at an eviction, the most recent access to the victim is selected to use MRU



OPT simulation

- This LRU-MRU cache is optimal [Gu et al., 2008]

UNIVERSITY of ROCHESTER

PACMAN

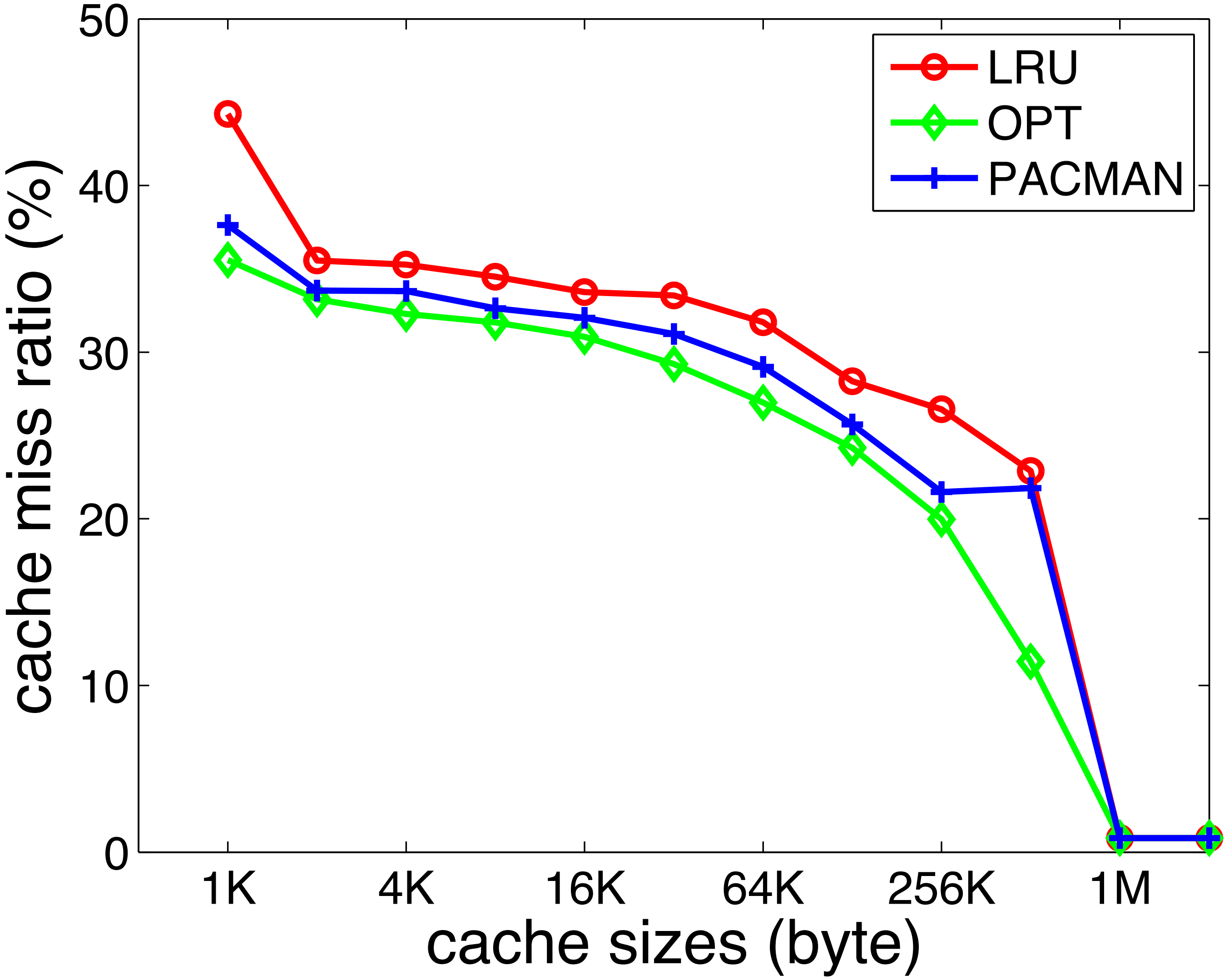
- Program-assisted Cache Management
  - do LRU-MRU collaborative caching at program level
    - restriction: run-time accesses from the same static memory reference must use the same access type
  - a simple model to select static memory references to use MRU
    - based on the optimal LRU-MRU selection
    - a reference has an **MRU ratio** of x if x fraction of accesses by this reference are selected to use MRU in the optimal LRU-MRU selection
    - select a static memory reference to use MRU if <u>MRU ratio ≥ 50%</u>

Testing Configurations

- Collect memory traces
  - do instrumentation in LLVM
- Cache simulator
  - single-level fully-associative cache
  - cache line size: 8 bytes
  - cache sizes: from 1KB to the double size of data set

UNIVERSITY of ROCHESTER

173.applu



OPT imprv.(C) =

$$\frac{miss_{LRU}(C) - miss_{OPT}(C)}{miss_{LRU}(C)}$$

PACMAN imprv.(C) =

$$\frac{miss_{LRU}(C) - miss_{PACMAN}(C)}{miss_{LRU}(C)}$$

- Avg. OPT imprv.: 17%
- Avg. PACMAN imprv.: 8.2%

UNIVERSITY of ROCHESTER

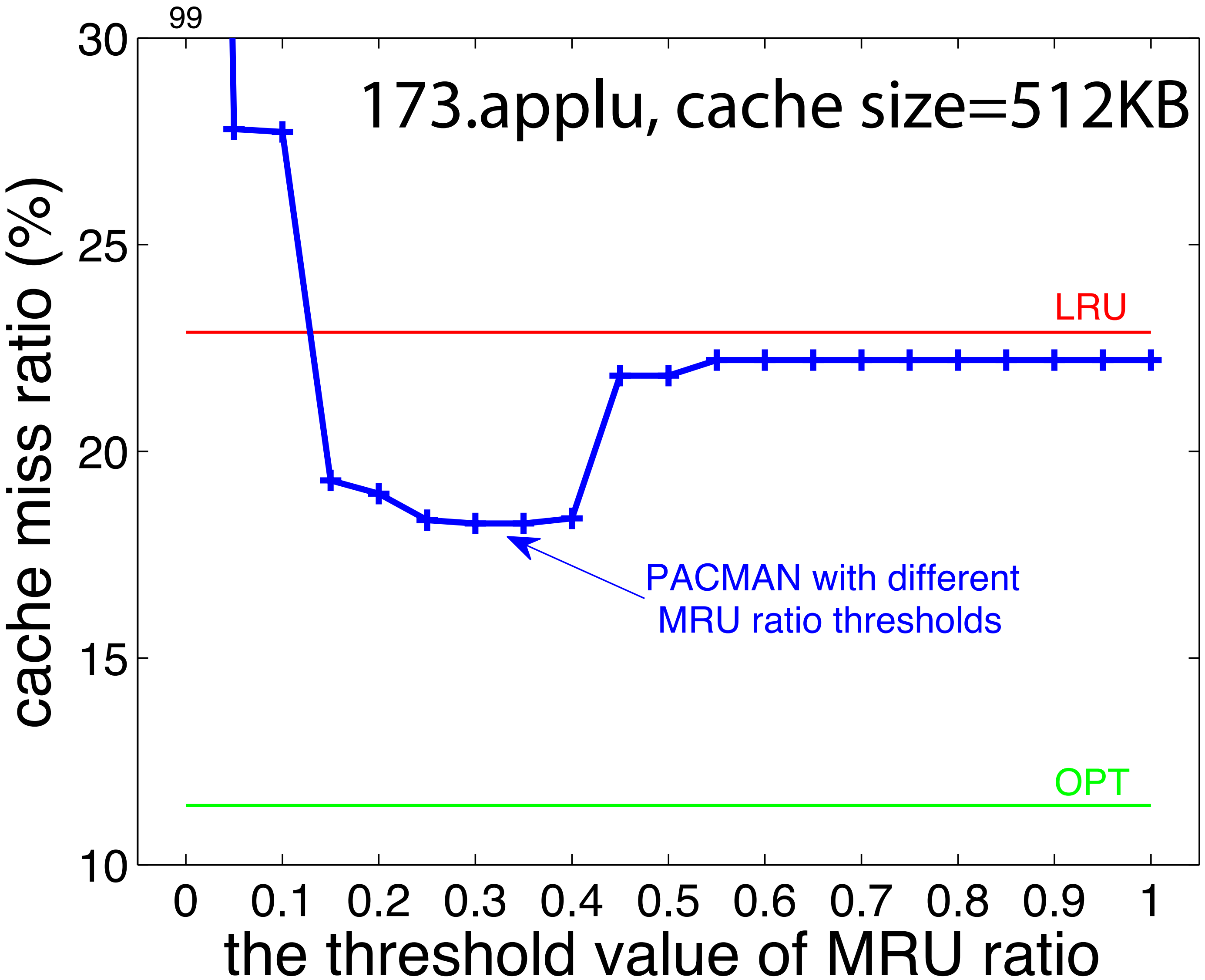
Overall Results

	the OPT imprv. over LRU		the PACMAN imprv. over LRU	
	average	largest	average	largest
SOR	25%	91%	15%	91%
171.swim	19%	64%	12%	59%
172.mgrid	31%	60%	13%	46%
173.applu	17%	50%	8.2%	19%
183.equake	22%	54%	17%	54%
189.lucas	34%	67%	26%	64%
410.bwaves	25%	80%	12%	60%
433.milc	31%	62%	8.8%	22%
434.zeusmp	12%	79%	1.4%	3.9%
437.leslie3d	27%	50%	10%	29%
average	24%	66%	12%	45%

- Half possible improvement is achieved in average

UNIVERSITY of ROCHESTER

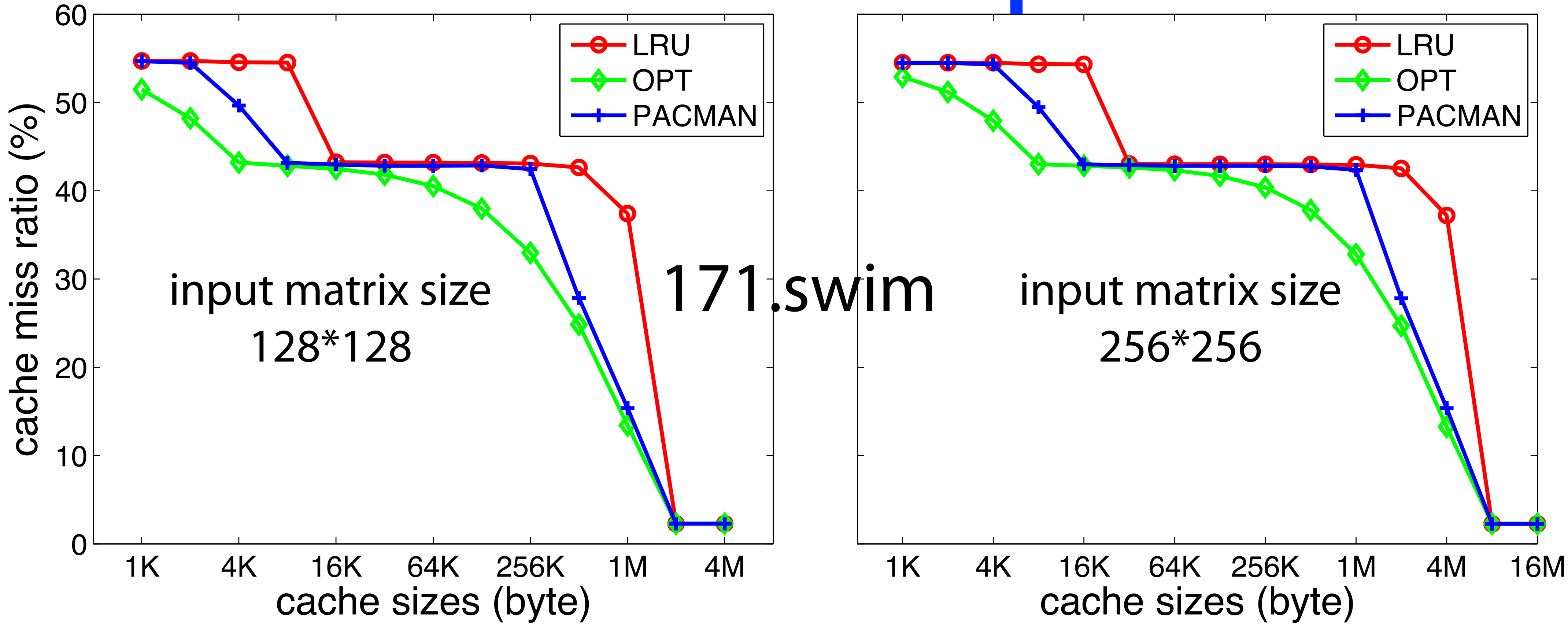
The Impact of MRU Ratio Threshold



- The threshold matters
  - PACMAN imprv.=4.6% if MRU ratio threshold=50%
  - PACMAN imprv.=20% if MRU ratio threshold=30% or 35%

UNIVERSITY of ROCHESTER

The Effect of Different Inputs



- Similar improvement showed with different inputs
  - possible to enable a feedback-based optimization from a training run with a smaller input

Summary

- LRU-MRU collaborative caching
  - holds inclusion property
    - an algorithm to compute the LRU-MRU stack distance
  - has promising potential
    - achieves half possible improvement with 10 benchmarks