Cache performance

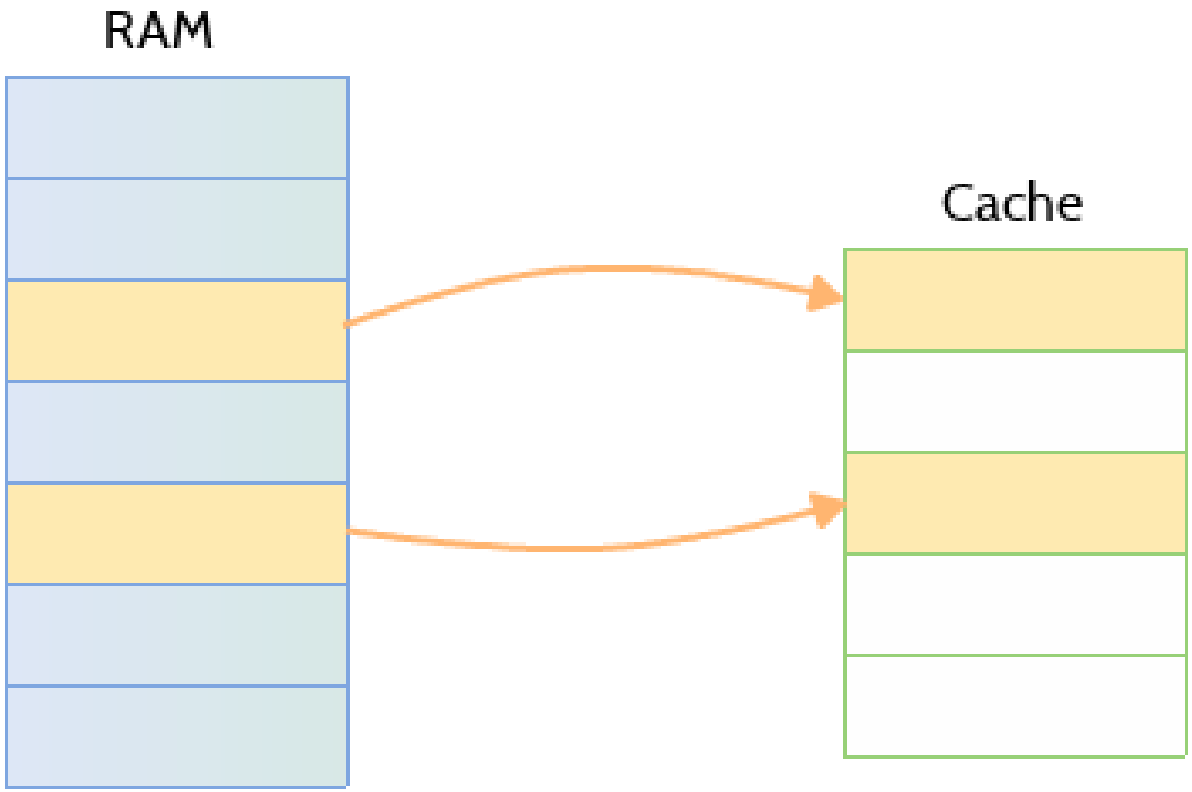
Le Nguyen Dung 7140224
Le Nguyen Khanh Duy 7140226
Pham Minh Thien 7140258

Ho Chi Minh City University of Technology

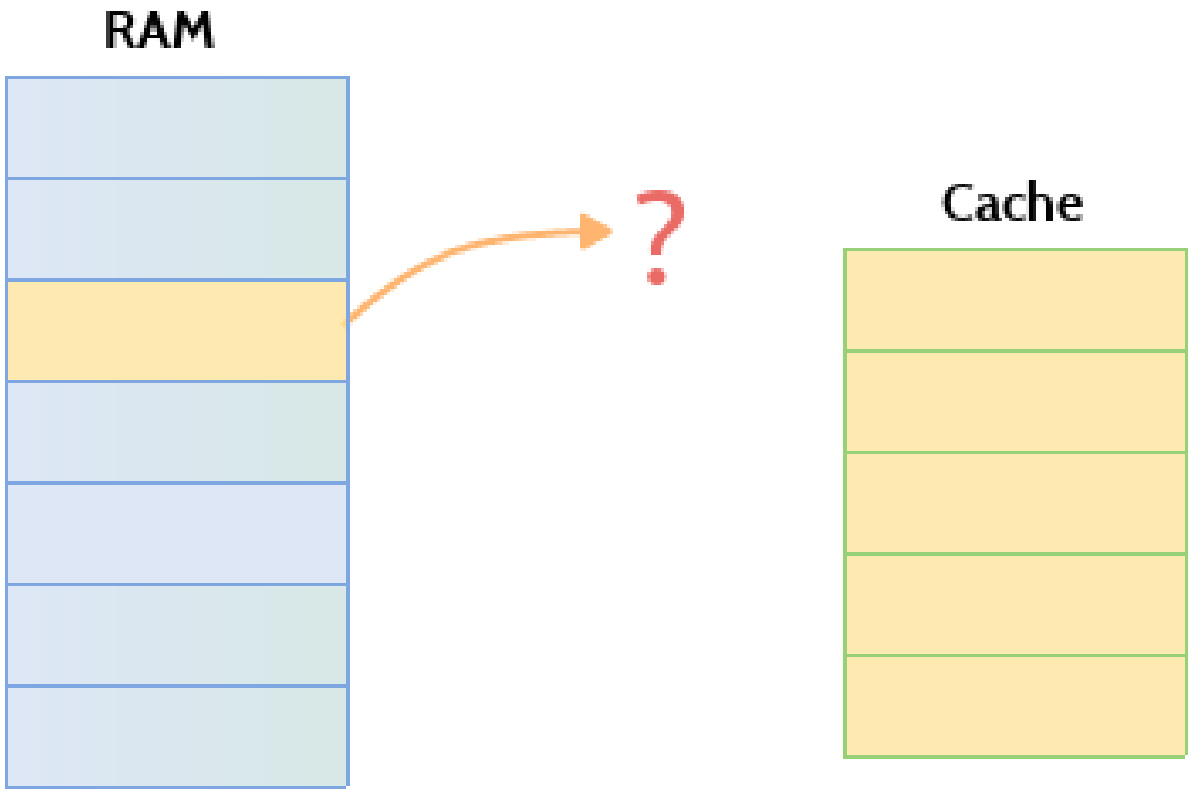
Instructor: Dr. Tran Ngoc Thinh

Content

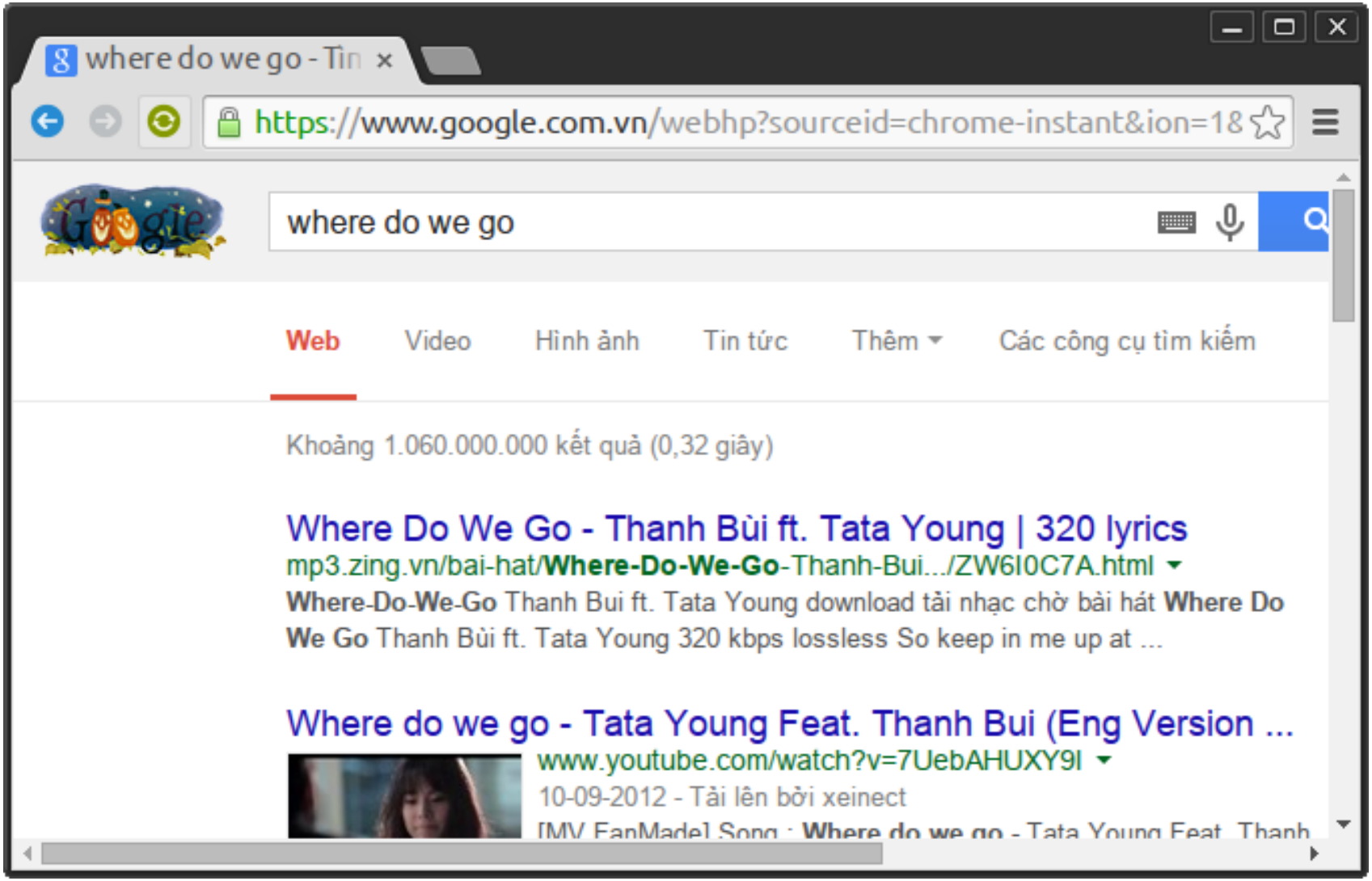
What is replacement policy ?



What is replacement policy ?



Where do cache go ?



Cache replacement save the world

Cache replacement

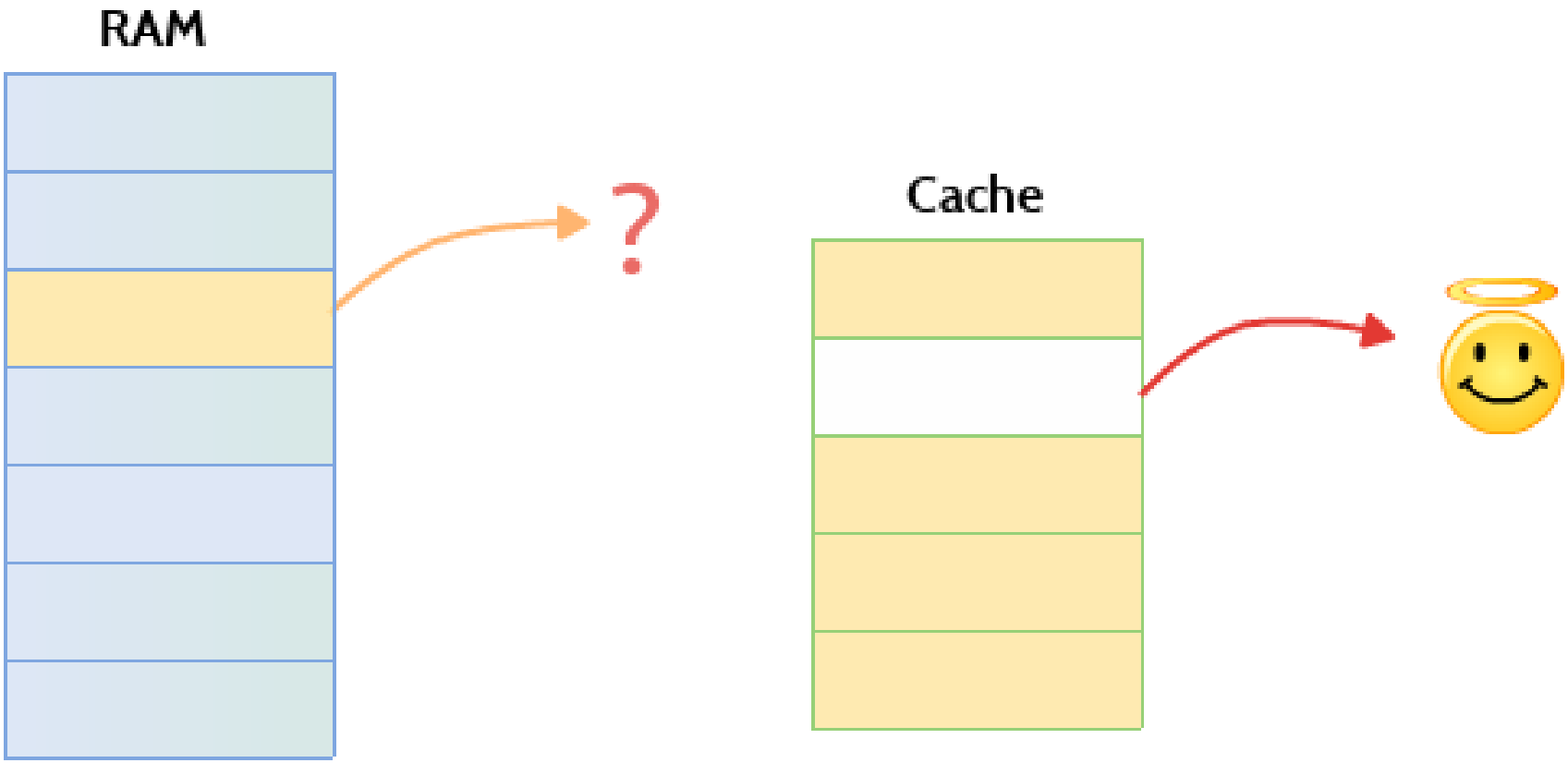


Figure 2 : Cache evict

Cache replacement

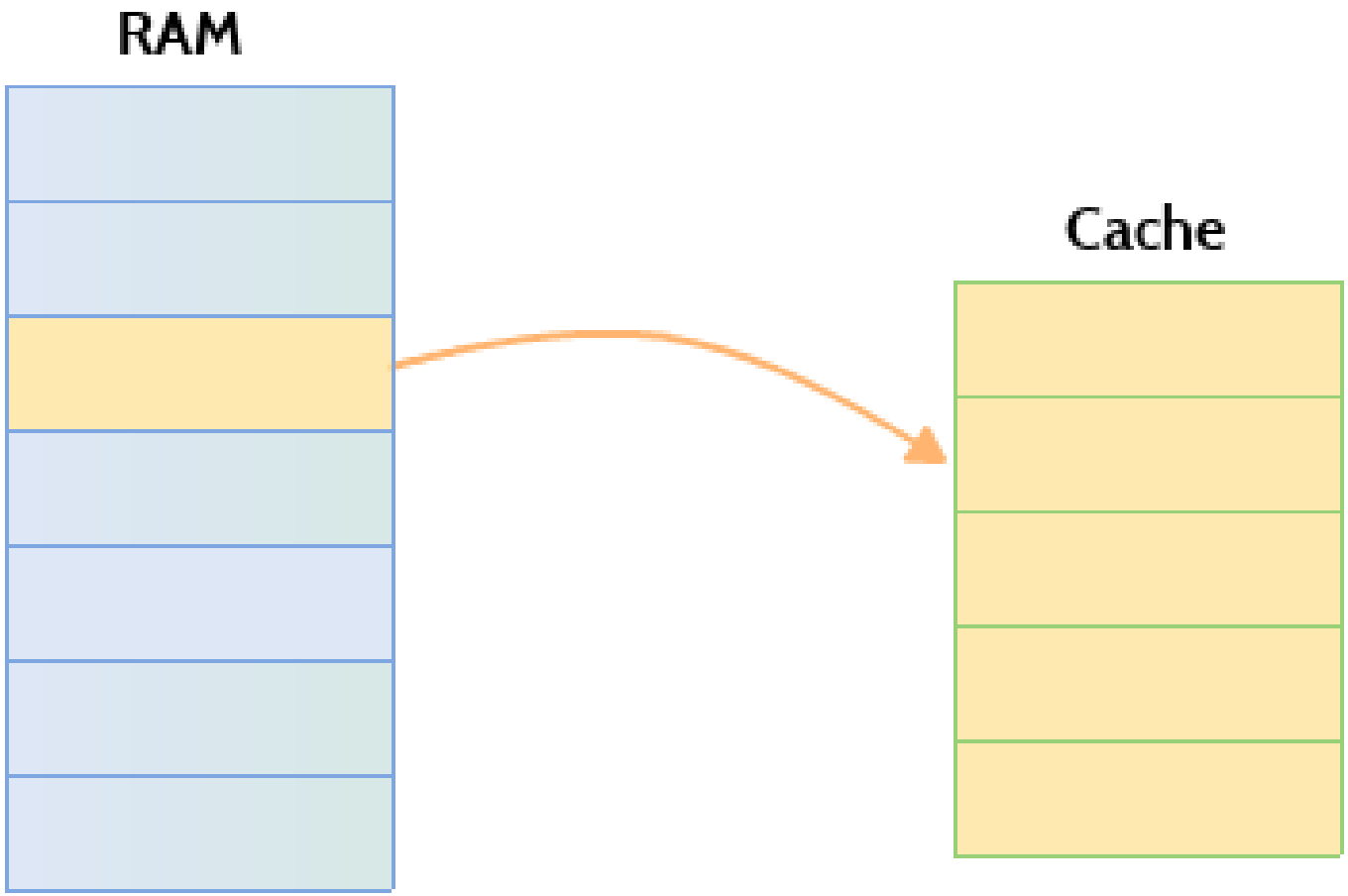


Figure 3 : Cache replacement

Replacement policy

- Not recently used
- First-in, first-out
- Least recently used
- Not frequently used
- Random

Not recently used

- Remove the page used the least often since the last clock cycle
  - 3. referenced, modified
  - 2. referenced, not modified
  - 1. not referenced, modified
  - 0. not referenced, not modified

First-in, first-out

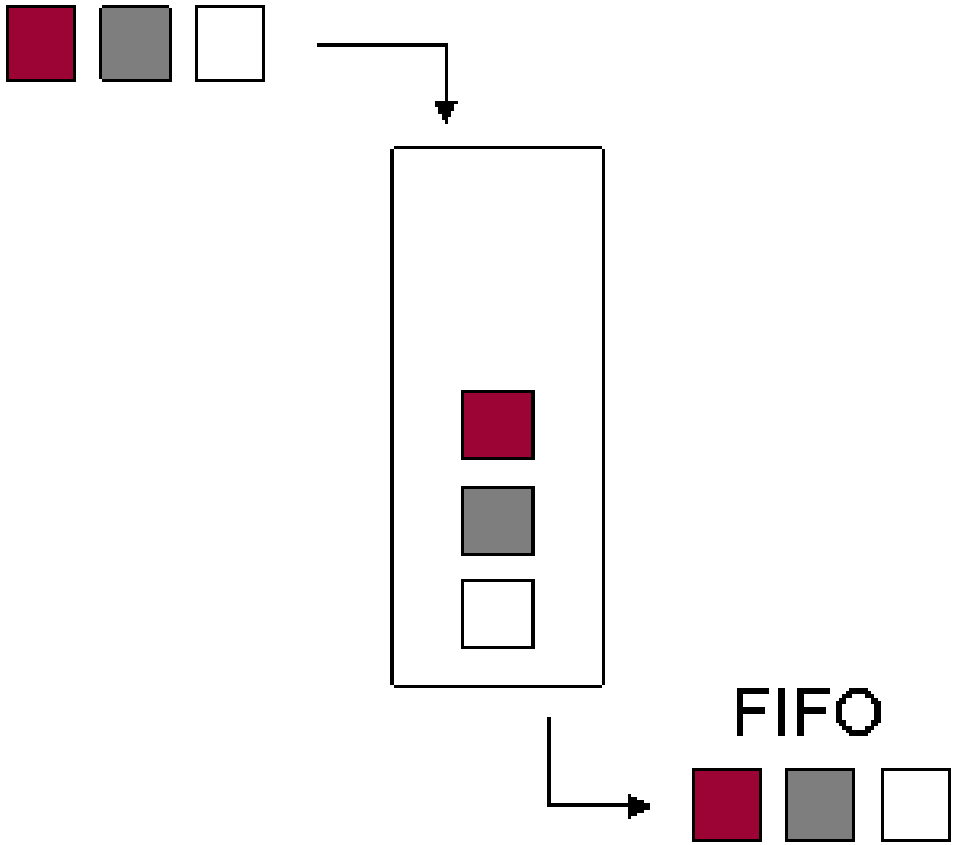
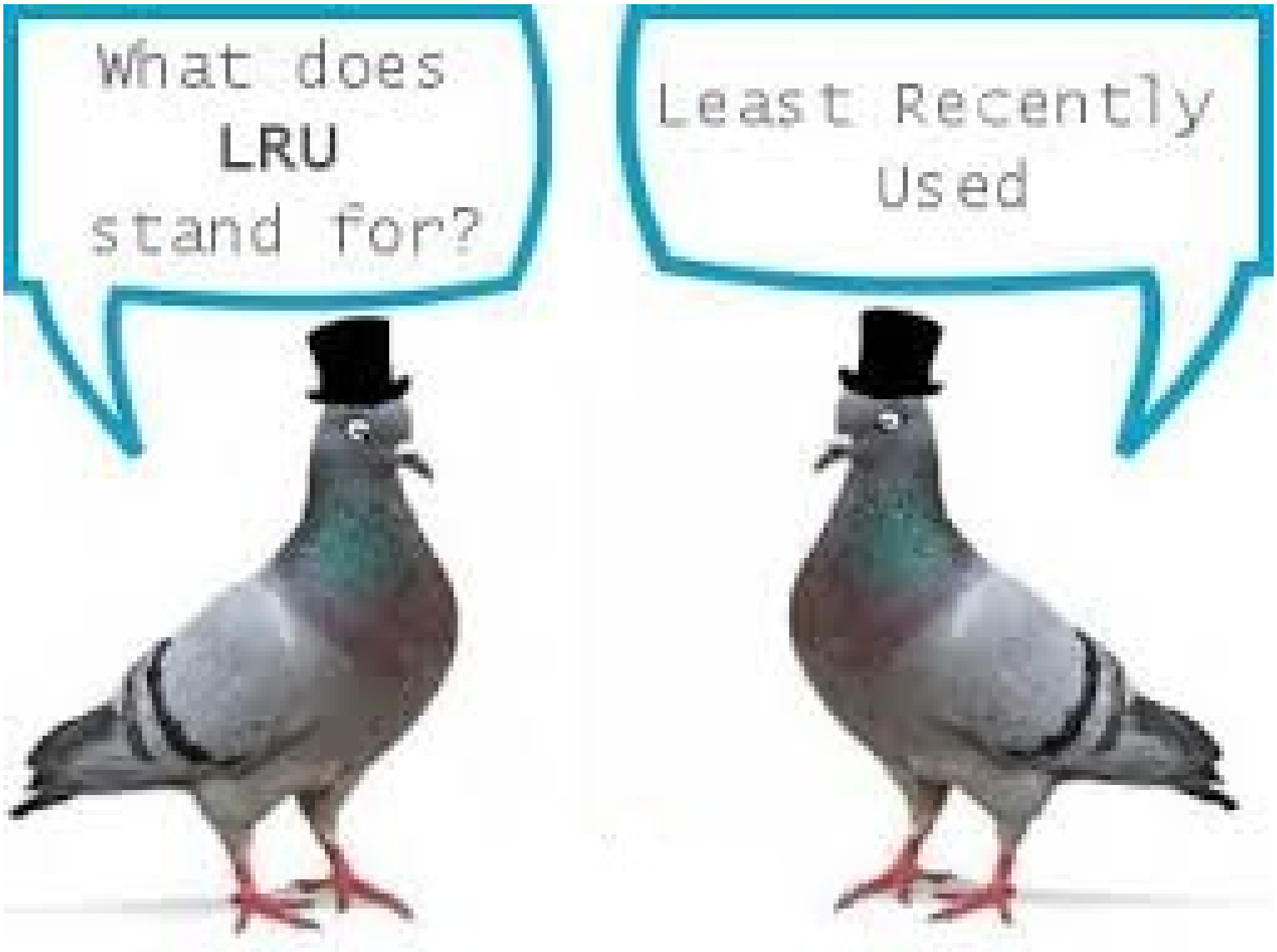


Figure 4 : FIFO

Least recently used



Not frequently used

- The not frequently used (NFU) page replacement algorithm requires a counter, and every page has one counter of its own which is initially set to 0.
- each clock interval, all pages that have been referenced within that interval will have their counter incremented by 1. In effect, the counters keep track of how frequently a page has been used. Thus, the page with the lowest counter can be swapped out when necessary.

Random Replacement (RR)

- Randomly selects a candidate item and discards it to make space when necessary.
- This algorithm does not require keeping any information about the access history.
- For its simplicity, it has been used in ARM processors It admits efficient stochastic simulation.

Optimization

- A better measure of memory hierarchy performance is the average memory access time:
  - Average memory access time = Hit time + Miss rate x Miss penalty

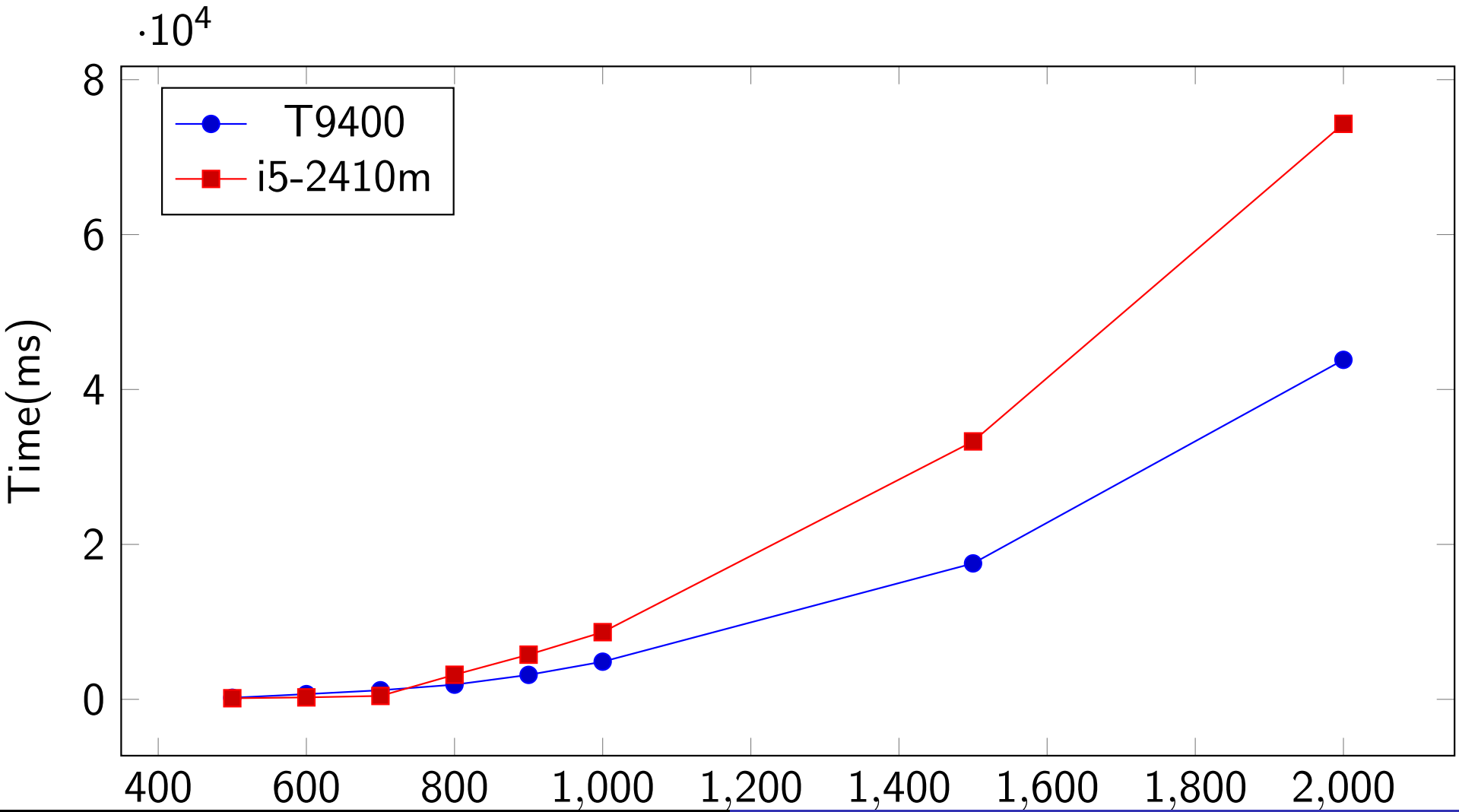
Optimization

- First Optimization: Larger Block Size to Reduce Miss Rate

	Cache size			
Block size	4K	16K	64K	256K
16	8.57%	3.94%	2.04%	1.09%
32	7.24%	2.87%	1.35%	0.70%
64	7.00%	2.64%	1.06%	0.51%
128	7.78%	2.77%	1.02%	0.49%
256	9.51%	3.29%	1.15%	0.49%

Optimization

- Second Optimization: Larger Caches to Reduce Miss Rate



Optimization

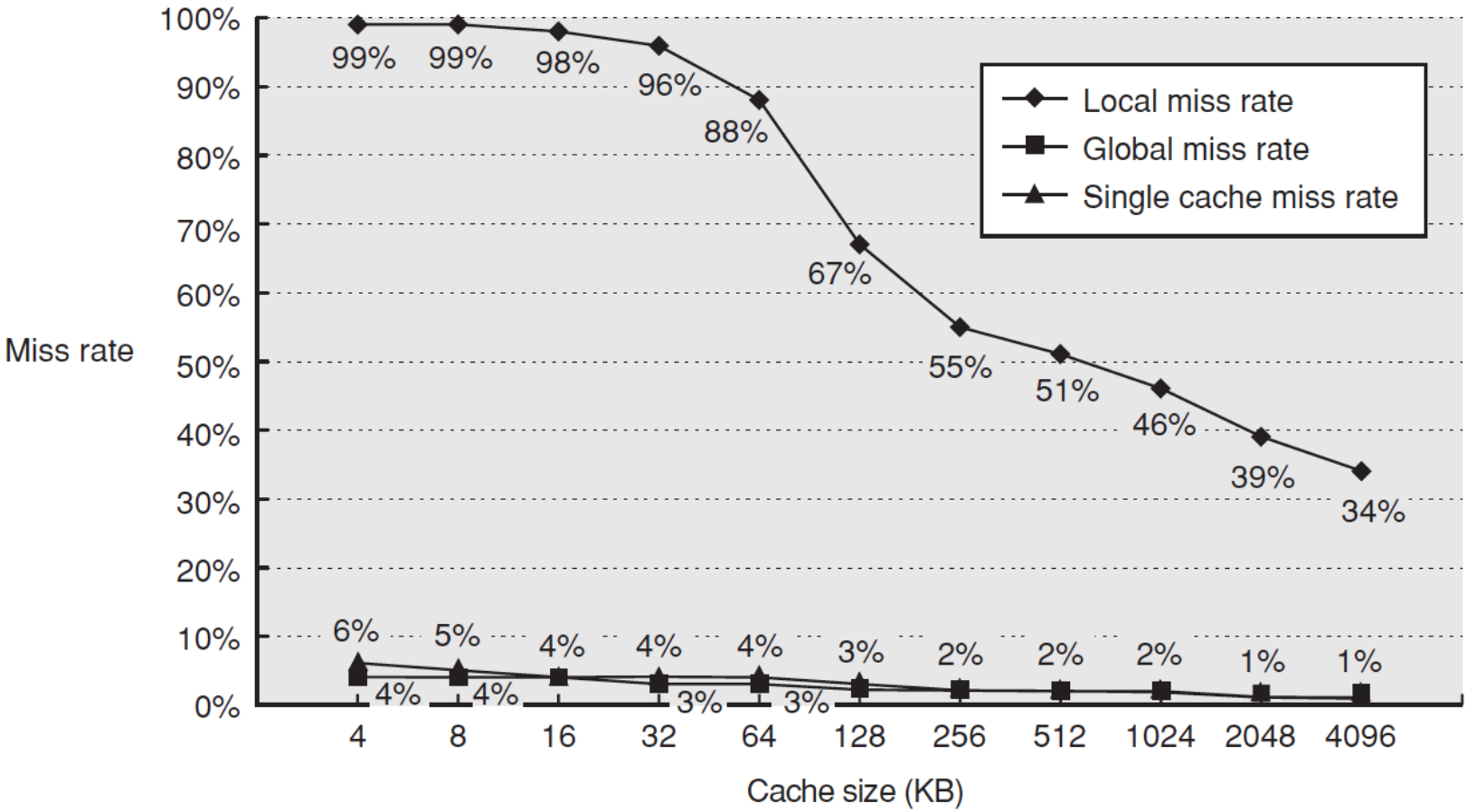
- Third Optimization: Higher Associativity to Reduce Miss Rate

	Associativity			
Cache size (KB)	1-way	2-way	4-way	8-way
4	3.44	3.25	3.22	**3.28**
8	2.69	2.58	2.55	**2.62**
16	2.23	**2.40**	**2.46**	**2.53**
32	2.06	**2.30**	**2.37**	**2.45**
64	1.92	**2.14**	**2.18**	**2.25**
128	1.52	**1.84**	**1.92**	**2.00**
256	1.32	**1.66**	**1.74**	**1.82**
512	1.20	**1.55**	**1.59**	**1.66**

Average memory access

Optimization

- Fourth Optimization: Multilevel Caches to Reduce Miss Penalty

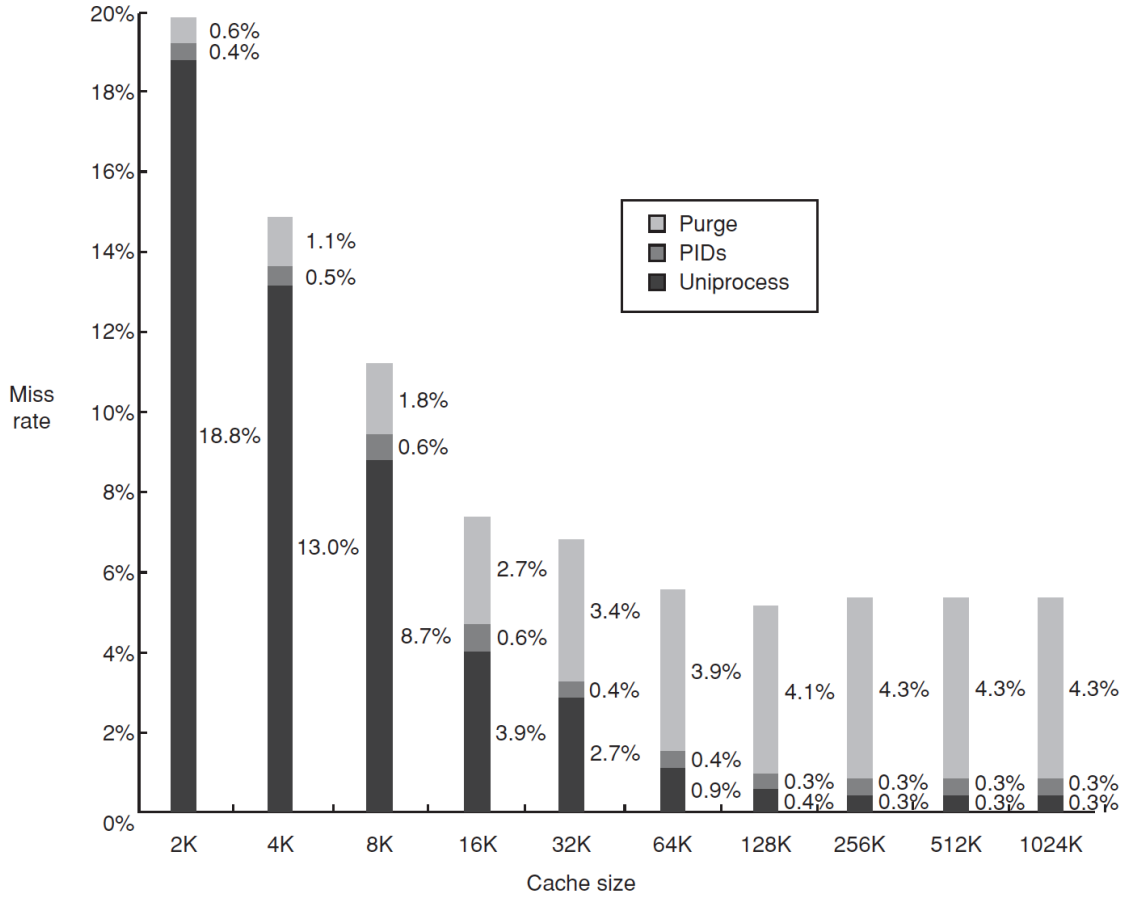


Optimization

- Fifth Optimization: Giving Priority to Read Misses over Writes to Reduce Miss Penalty

Optimization

- Sixth Optimization: Avoiding Address Translation during Indexing of the Cache to Reduce Hit Time



References

- Computer Architecture: A Quantitative Approach
- `http://en.wikipedia.org/wiki/Cache_algorithms`