CS698Y: Modern Memory Systems
Lecture-5 (Caches)

Biswabandan Panda

biswap@cse.iitk.ac.in

https://www.cse.iitk.ac.in/users/biswap/CS698Y.html

Flow of the Module

Cache Management Policies

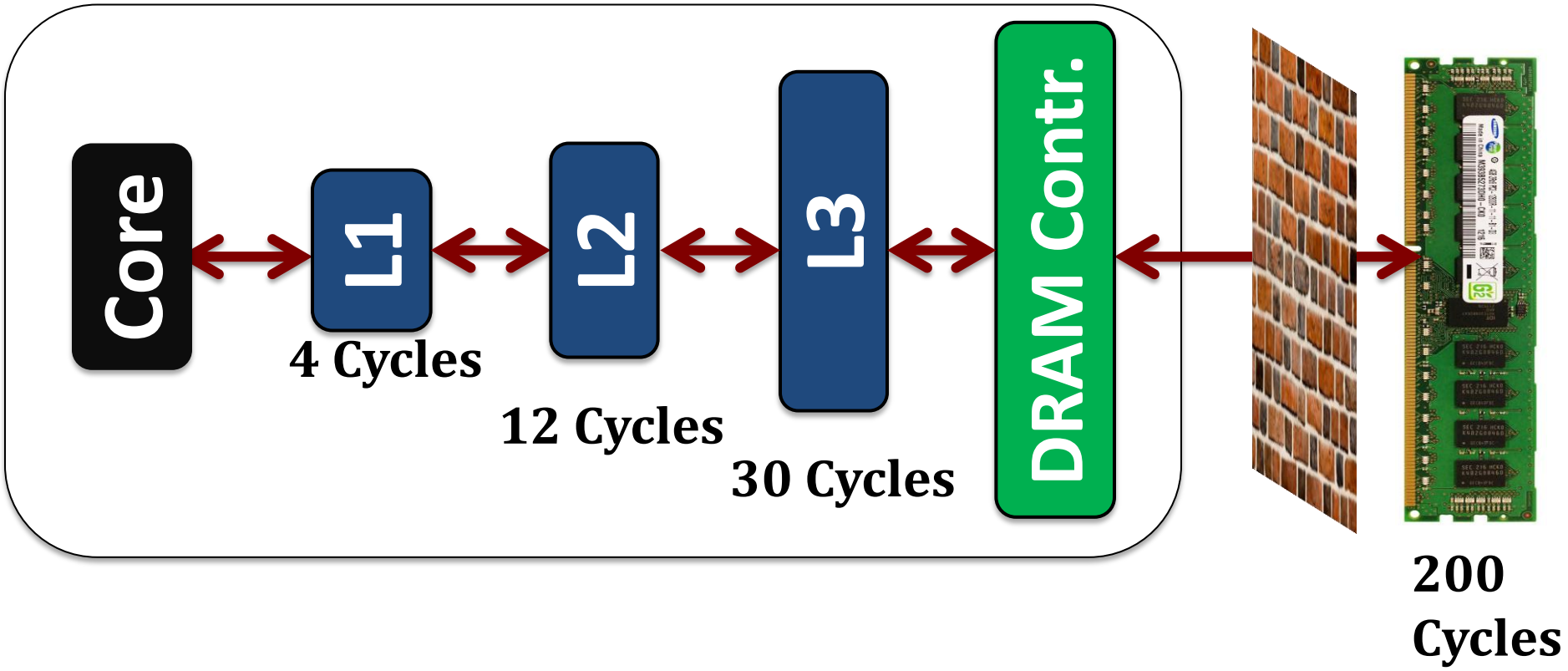
Cache Hierarchies

Hardware Prefetching

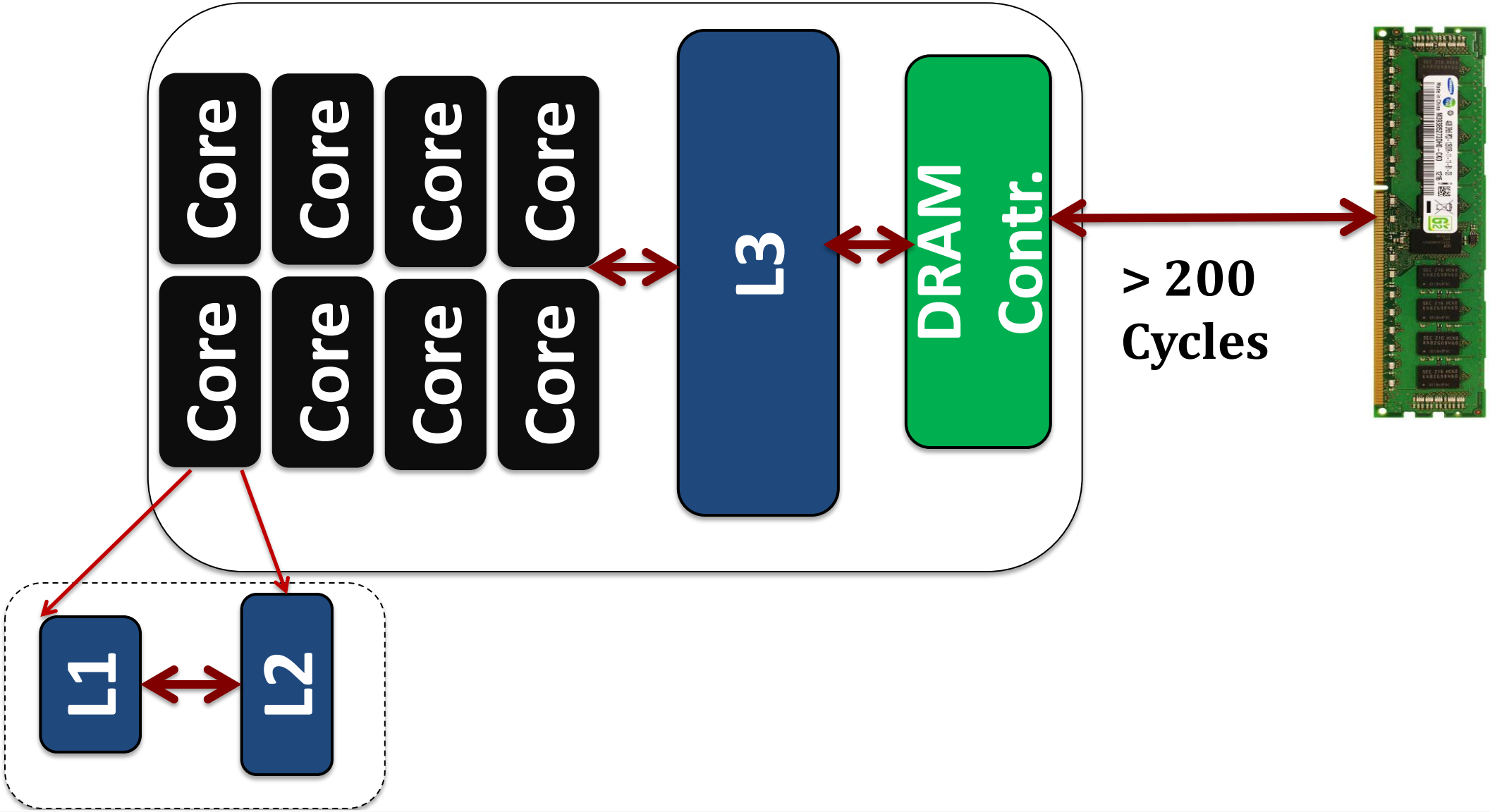
Cache Compression

Non-uniform Caches

Caches in Single-core System



Caches in Multi-core



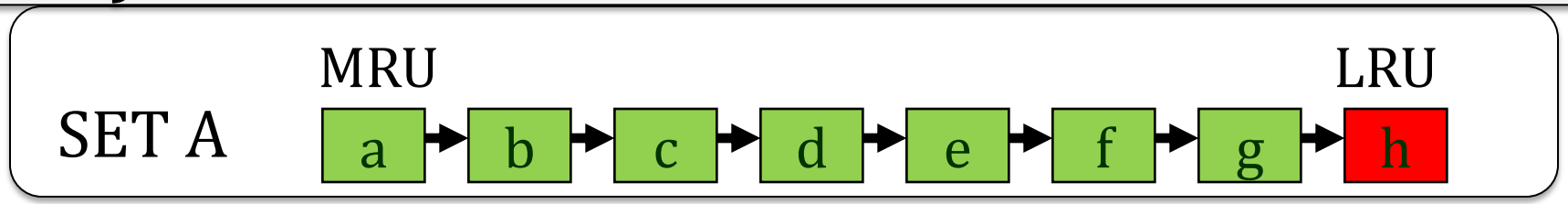
Latency Numbers

L1	Few Cycles
L2	Tens of Cycles
L3	Two to three times of L2
	Hundreds of cycles

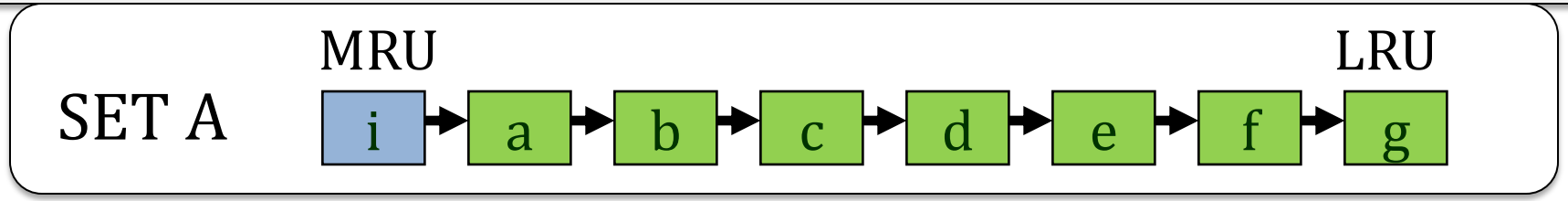
Our Goal:
To minimize off-chip DRAM accesses

Cache Replacement (LRU) - 101

Cache Eviction Policy: On a miss (block *i*), which block to evict (replace) ?



Cache Insertion Policy: New block *i* inserted into MRU.



Cache Promotion Policy: On a future hit (block *i*), promote to MRU

LRU causes thrashing when working set > cache size

Common Access Patterns [RRIP, ISCA 10]

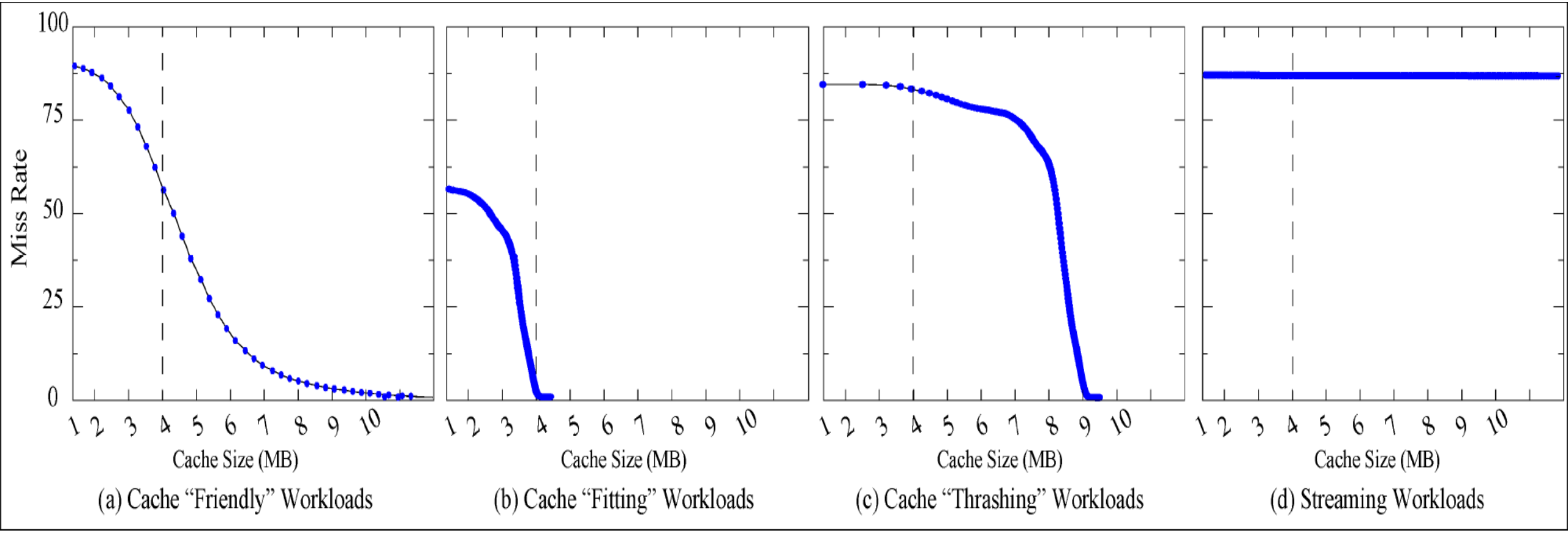
Recency friendly $(a_1, a_2, \ldots a_k, a_{k-1}, \ldots a_2, a_1)^N$

Thrashing $(a_1, a_2, \ldots a_k)^N$ [k > cache size]

Streaming $(a_1, a_2, \ldots a_\infty)^N$

Combination of above three

Types of Workloads (Baseline 4MB Cache)



Limitations of LRU

LRU exploits temporal locality

Streaming data (a_1, a_2, a_3,....a_∞):
No temporal locality,
No temporal reuse

Thrashing data (a_1, a_2, a_3,....a_n) [n>c]
Temporal locality exists. However, LRU fails to capture.

Bimodal Insertion Policy (BIP) [ISCA '07]

```
if  ( rand() < ε ) ε=1/16,1/32,1/64
    Insert at MRU position;
else
    Insert at LRU position;
```

For small ε: BIP retains thrashing protection of LRU insertion policy.

Infrequently insert lines in MRU position

Dynamic Insertion Policy (DIP) [ISCA '07]

SDM – Set Dueling monitors PSEL – n-bit saturating counters for deciding a policy