CS698Y: Modern Memory Systems
Lecture-10 (Cache Hierarchy)

Biswabandan Panda

biswap@cse.iitk.ac.in

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

Logistics

Quiz 1.0: Sept. 23, 13:00 to 15:00 hrs

P.A. 1: Sept. 10, 17:00 hrs, code + report @Canvas

Presentation: Sept. 14, 10:30 hrs, (9 + 1) mins

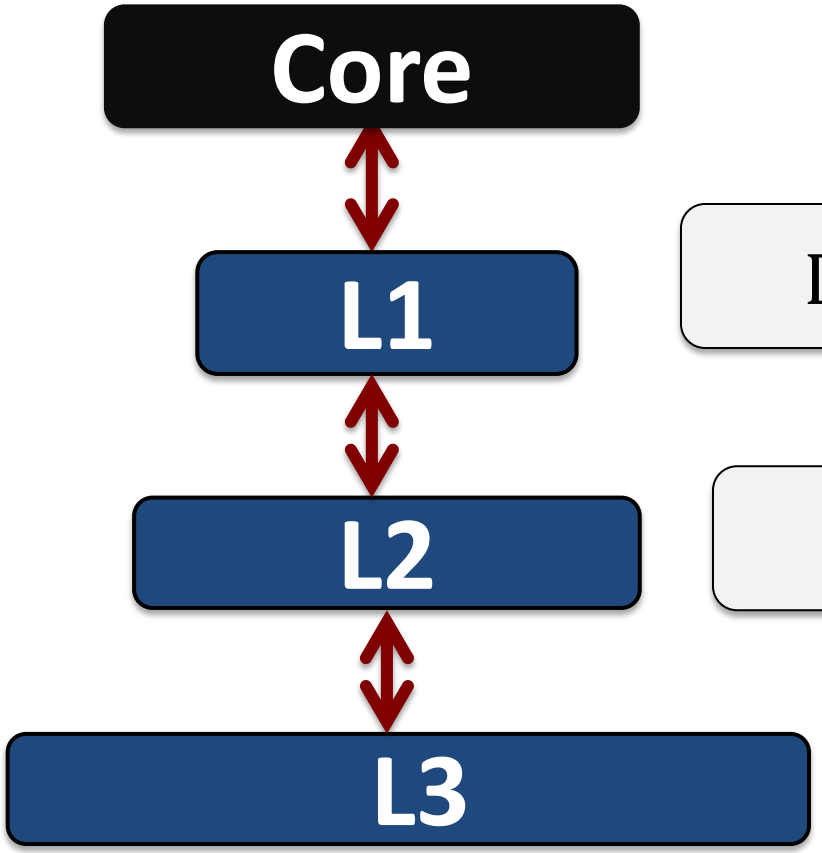
Insights: Why and how of what? For example:

What: astar, Hawkeye performs x% worse than LRU.

Why: x% of blocks have feature y, which is not captured by Hawkeye.

How: We can augment with a blackbox that does some black magic to improve Hawkeye.

Cache Hierarchy

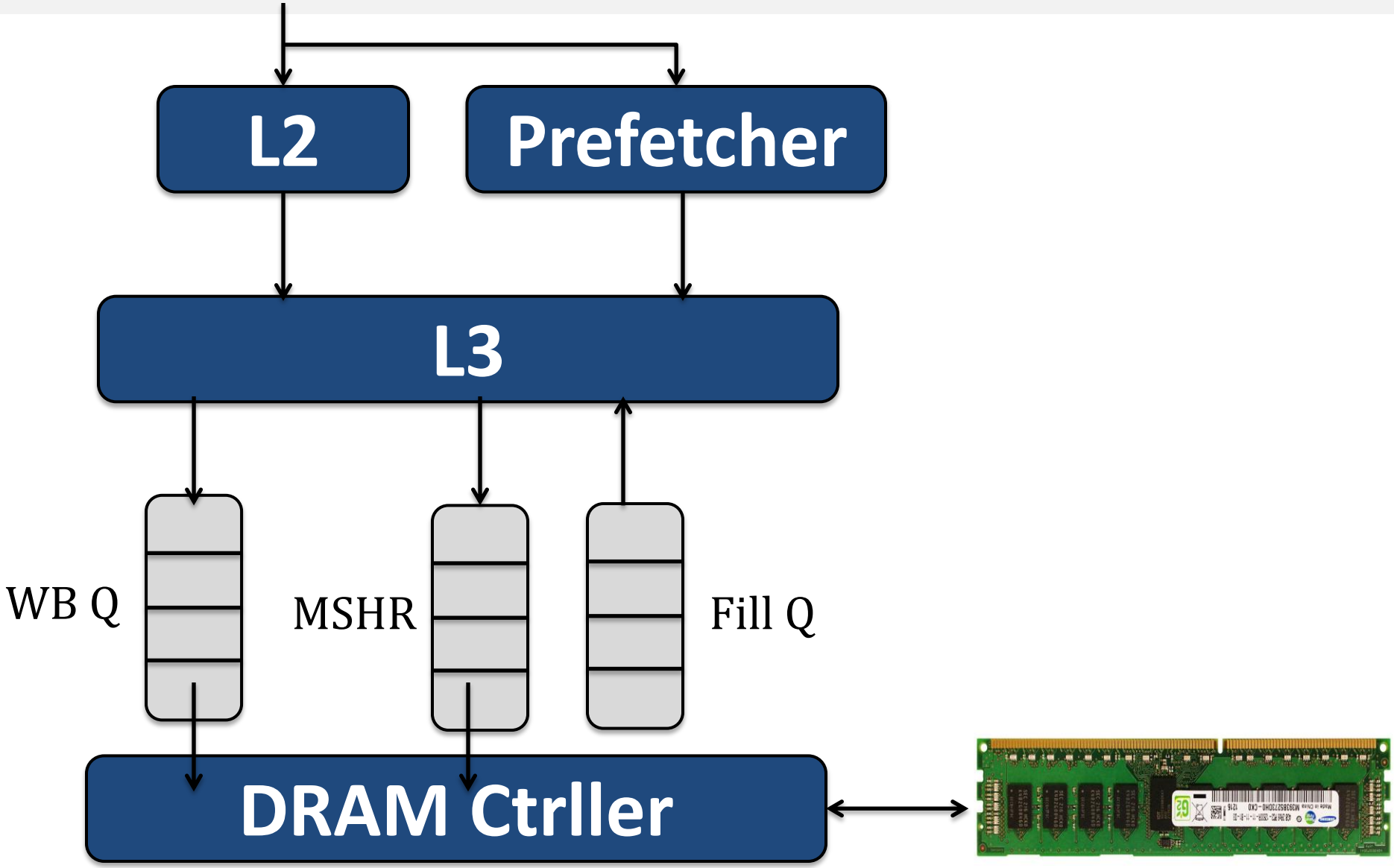


Designed for *bandwidth*

Designed for *latency*

Designed for *capacity*

Revisiting Cache Hierarchy (The Organization)

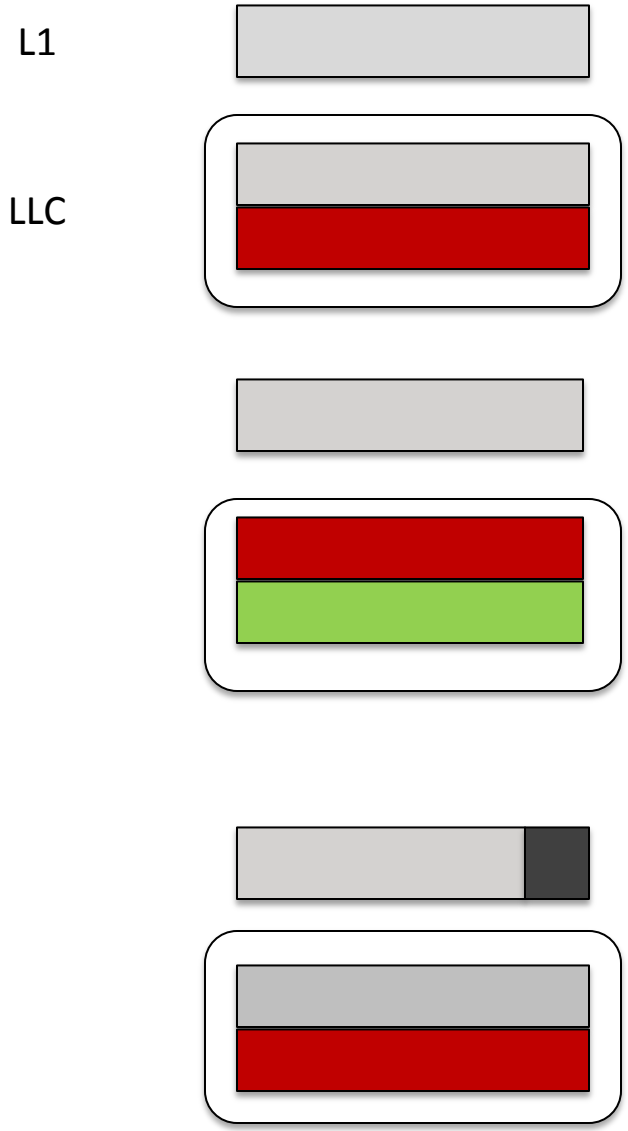


Cache Hierarchies

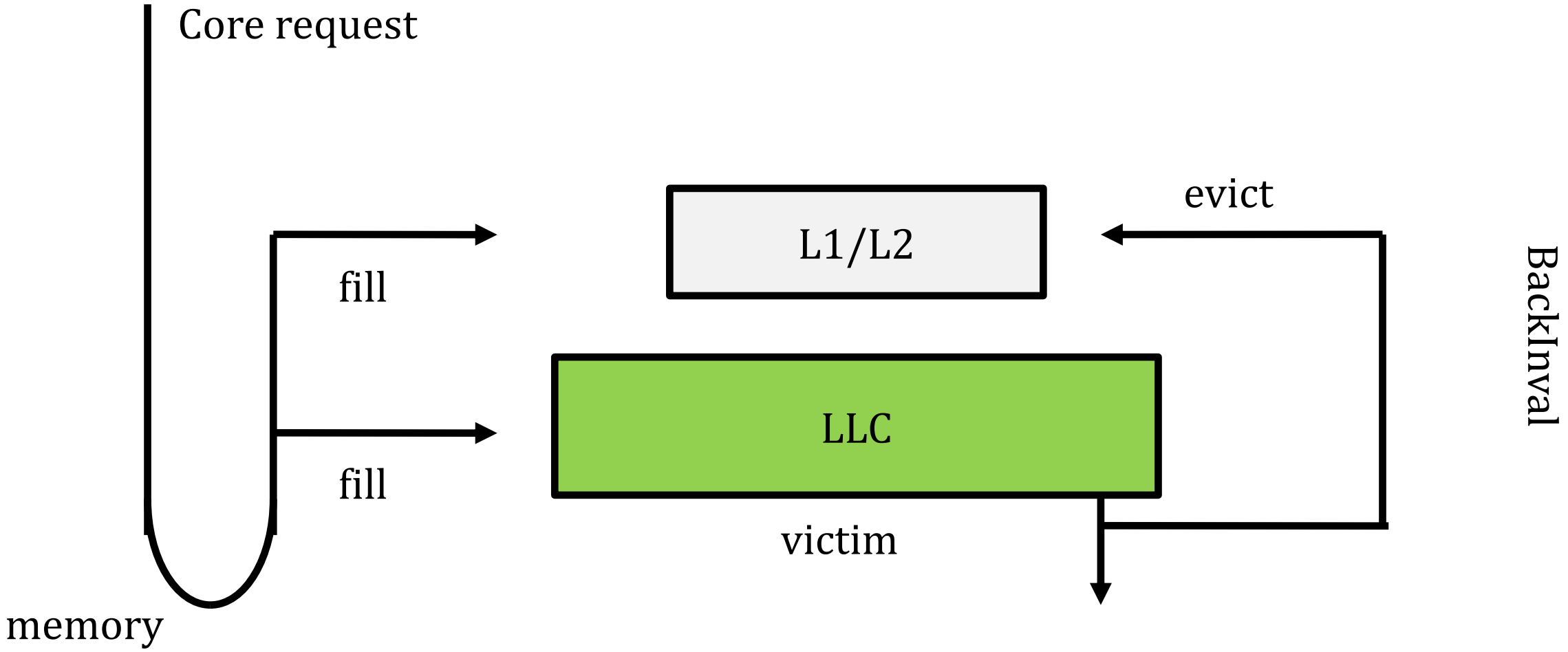
Inclusive: L1 cache blocks are present in L3

Exclusive: L1 cache blocks are absent in L3

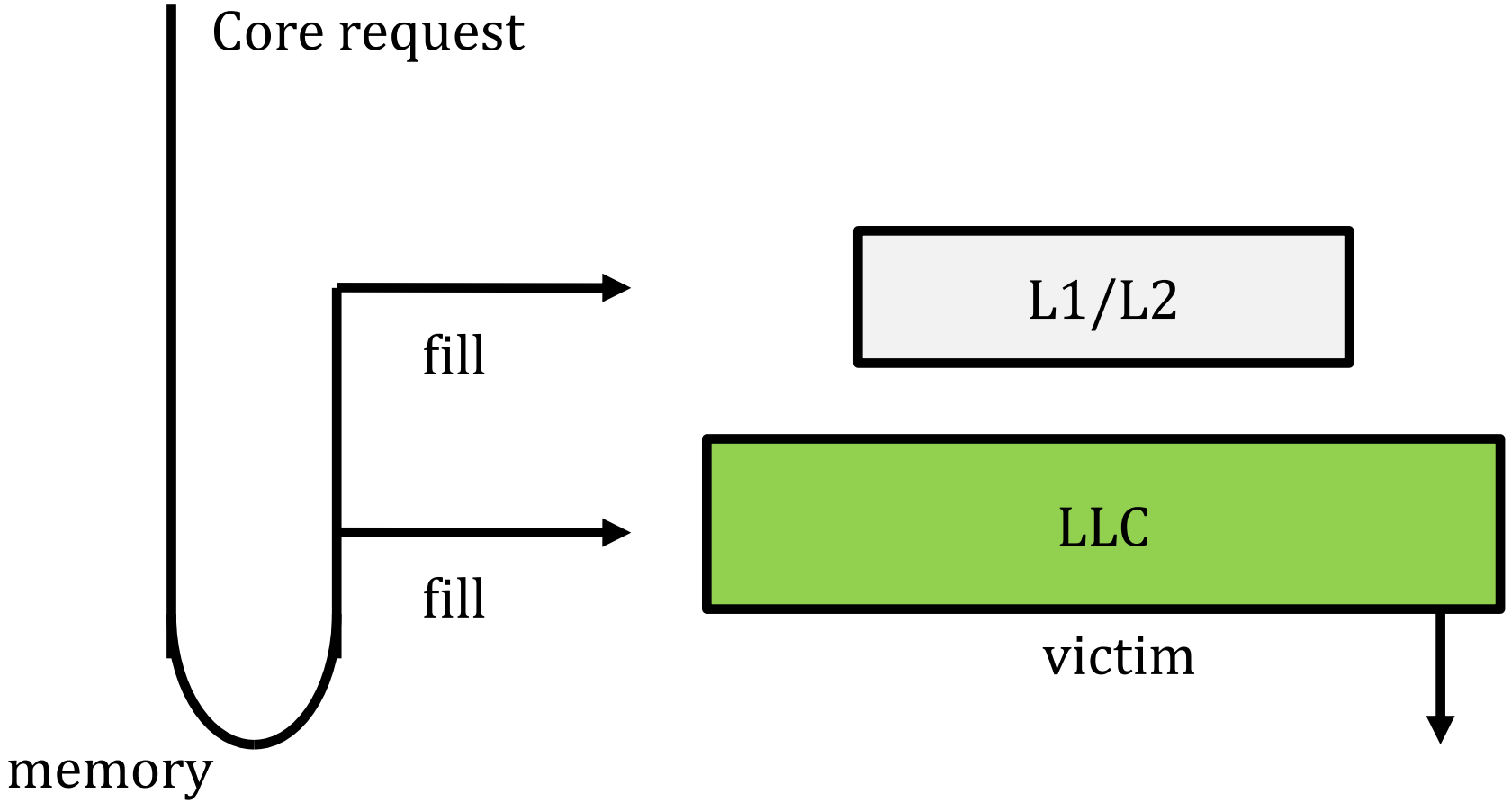
Non-inclusive: L1 cache blocks may/may not be in L3



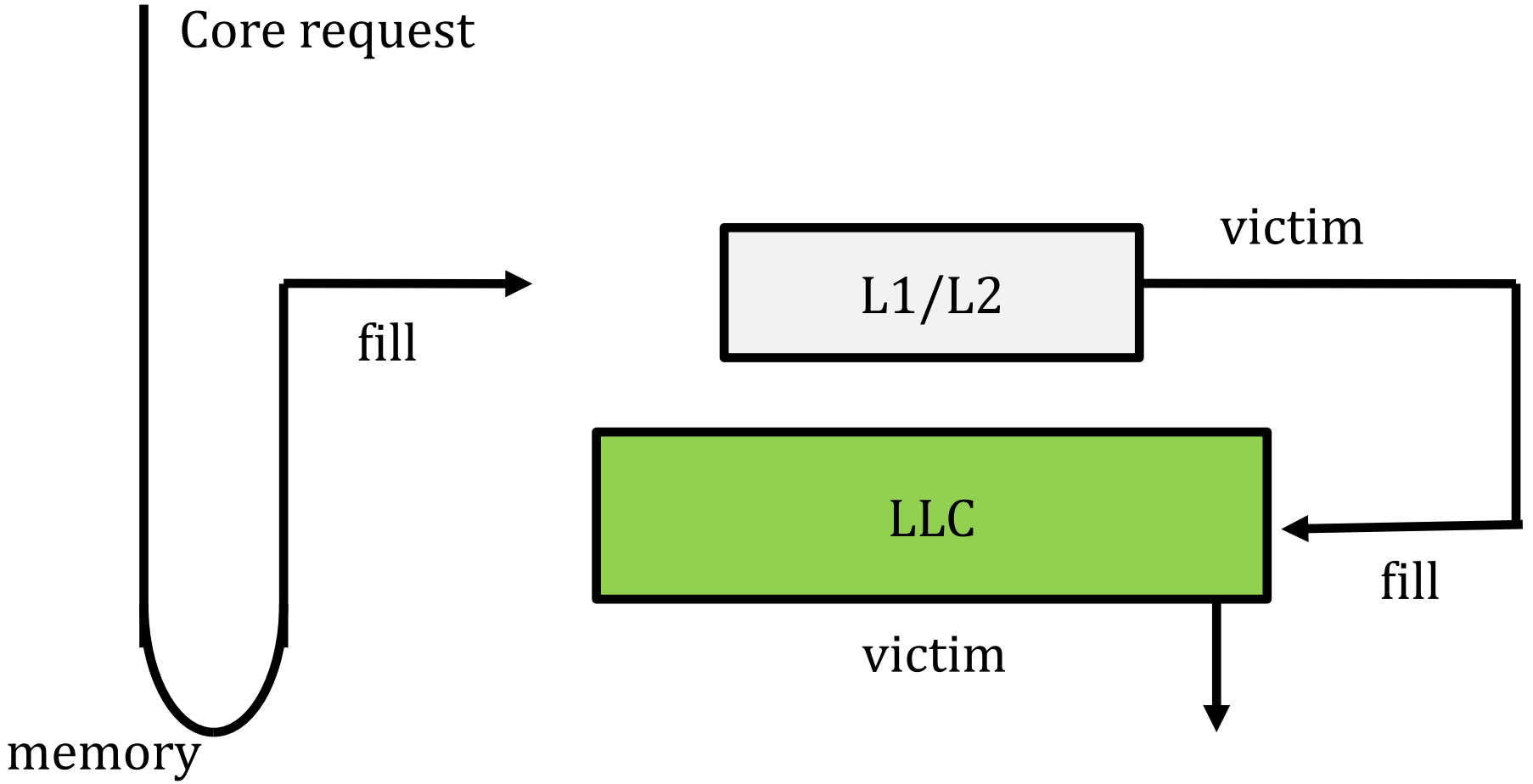
Inclusive Hierarchy



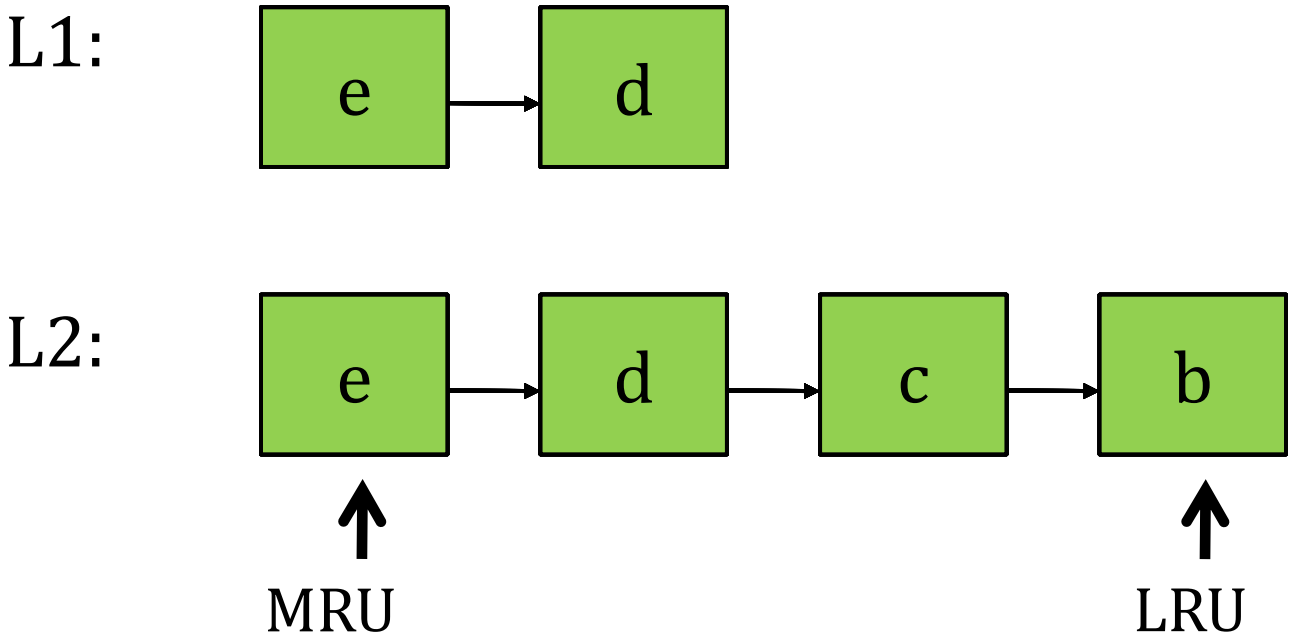
Non-inclusive



Exclusive Hierarchy



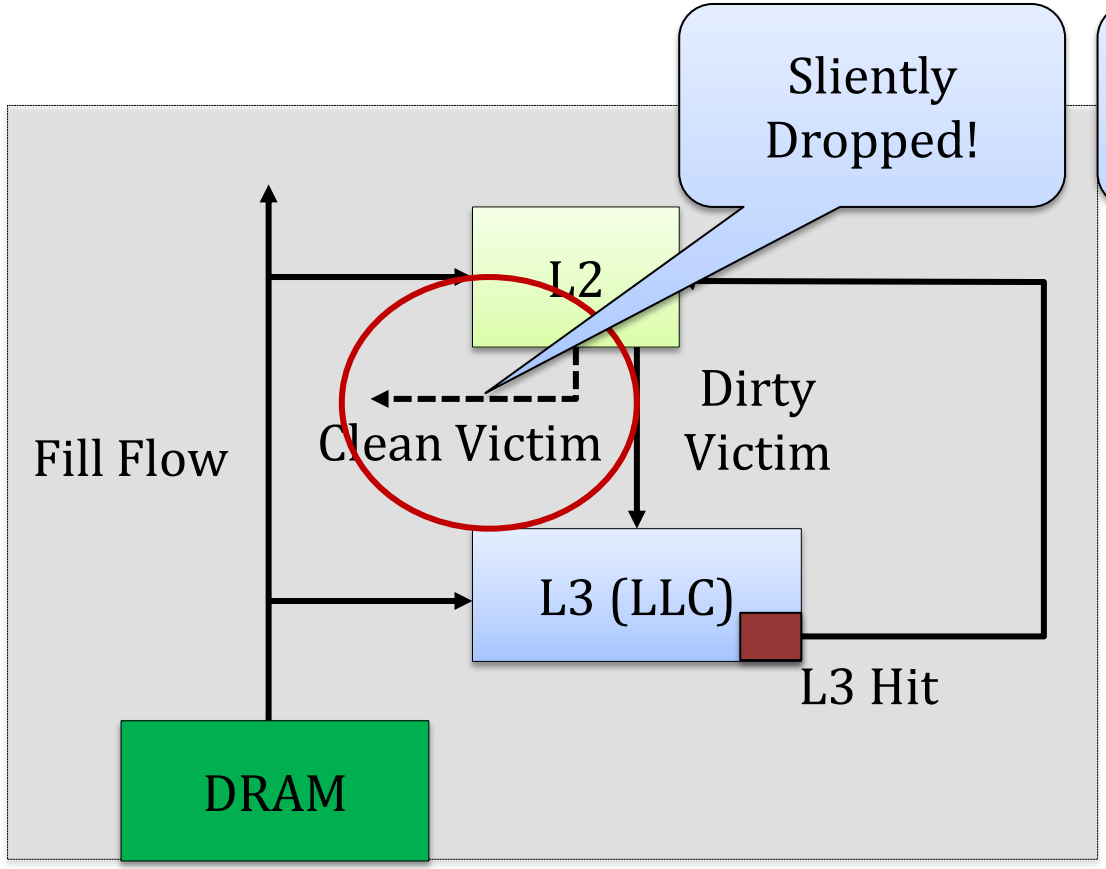
Back-invalidation [TLA, MICRO '10]



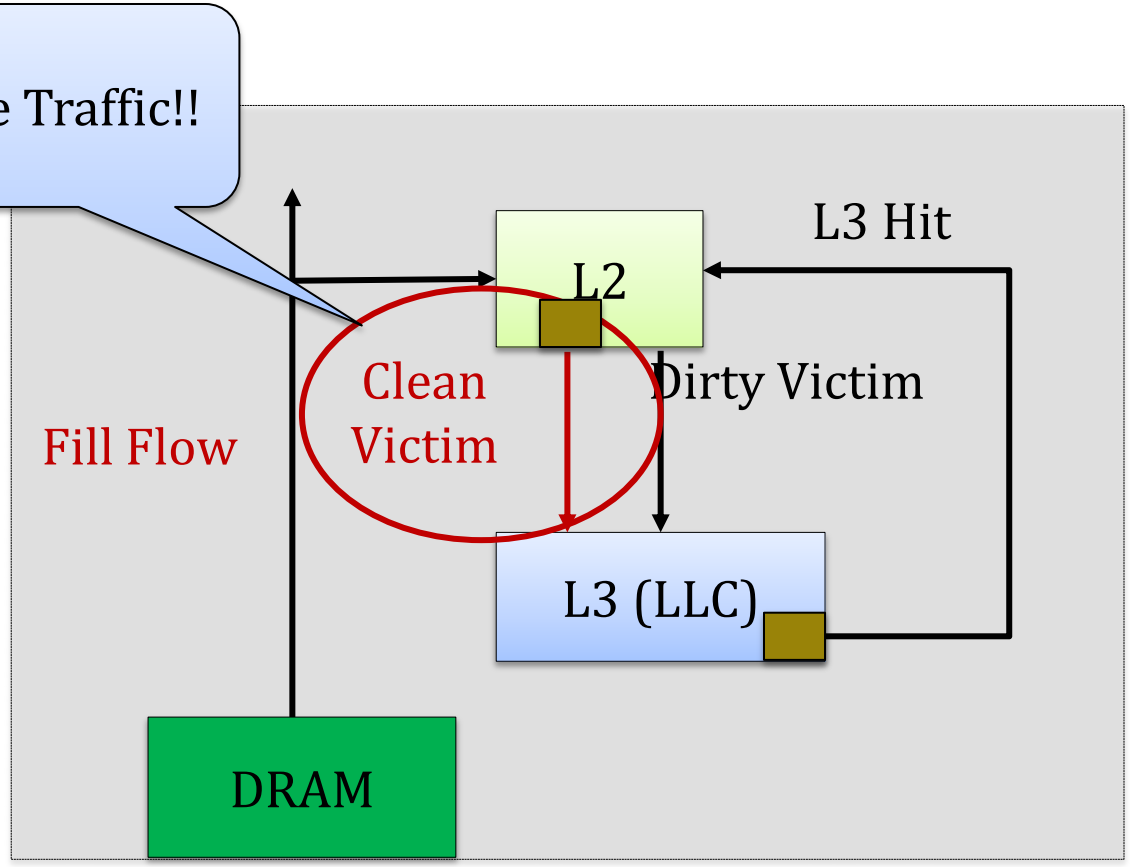
Reference 'e' misses and evicts 'a' from hierarchy

Next Reference to 'a' misses ☹

On-chip Traffic?

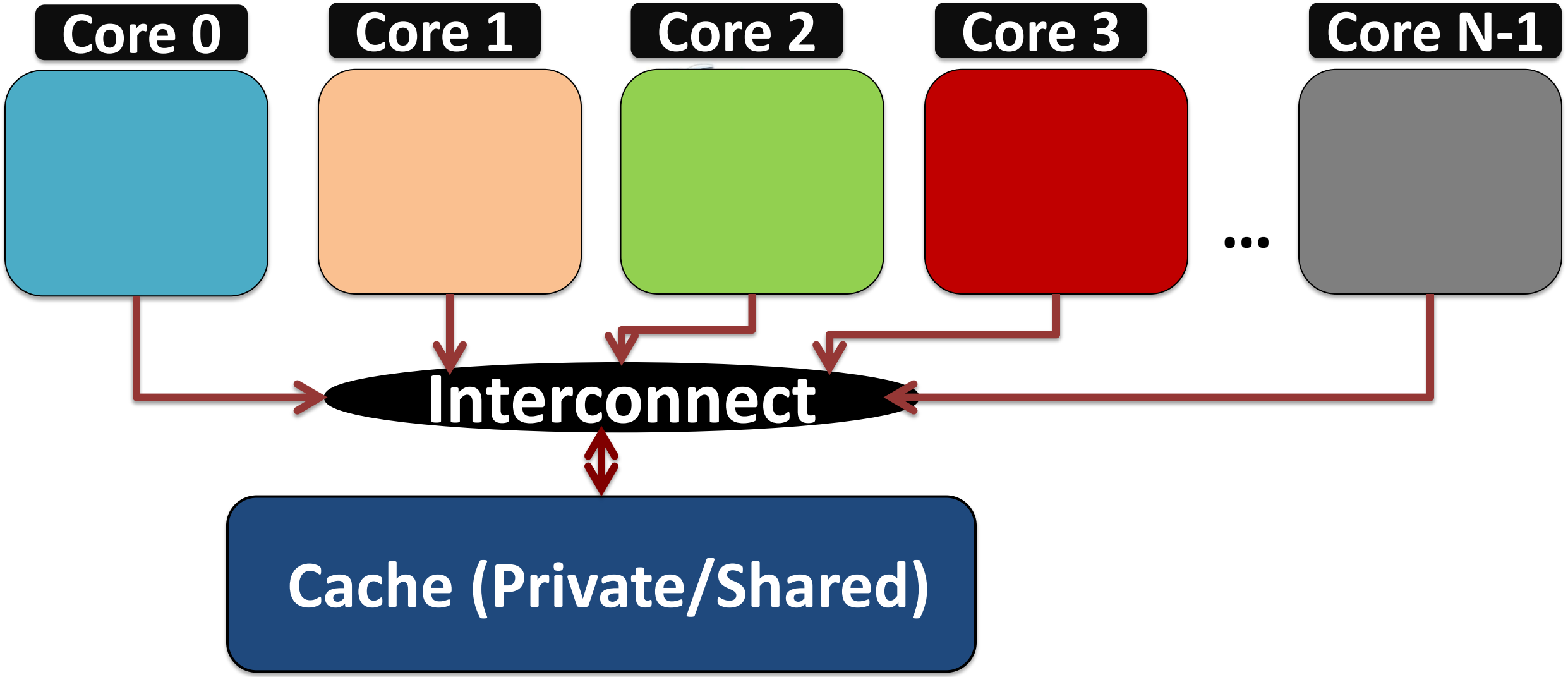


Non-Inclusive Hierarchy

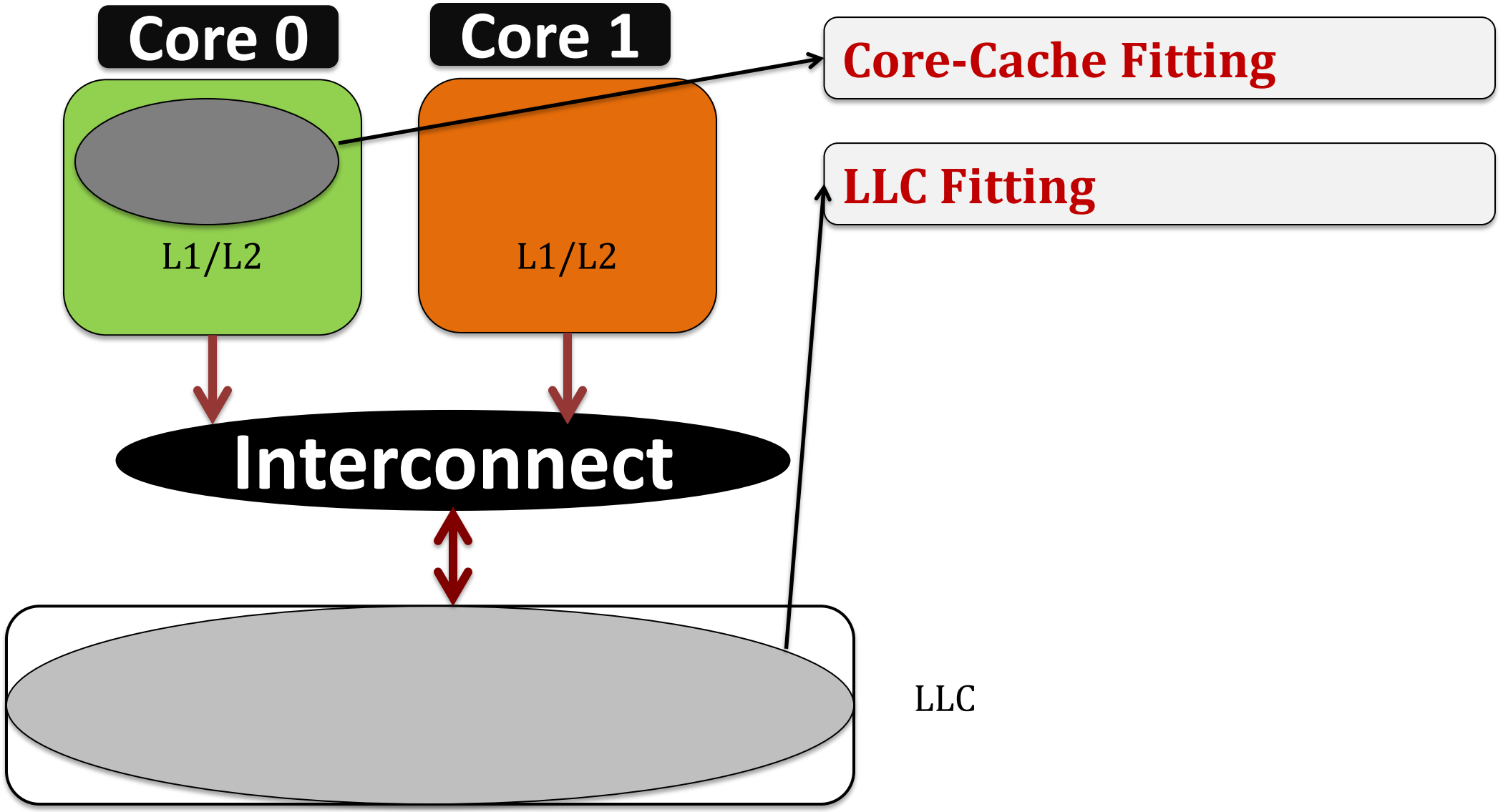


Exclusive Hierarchy

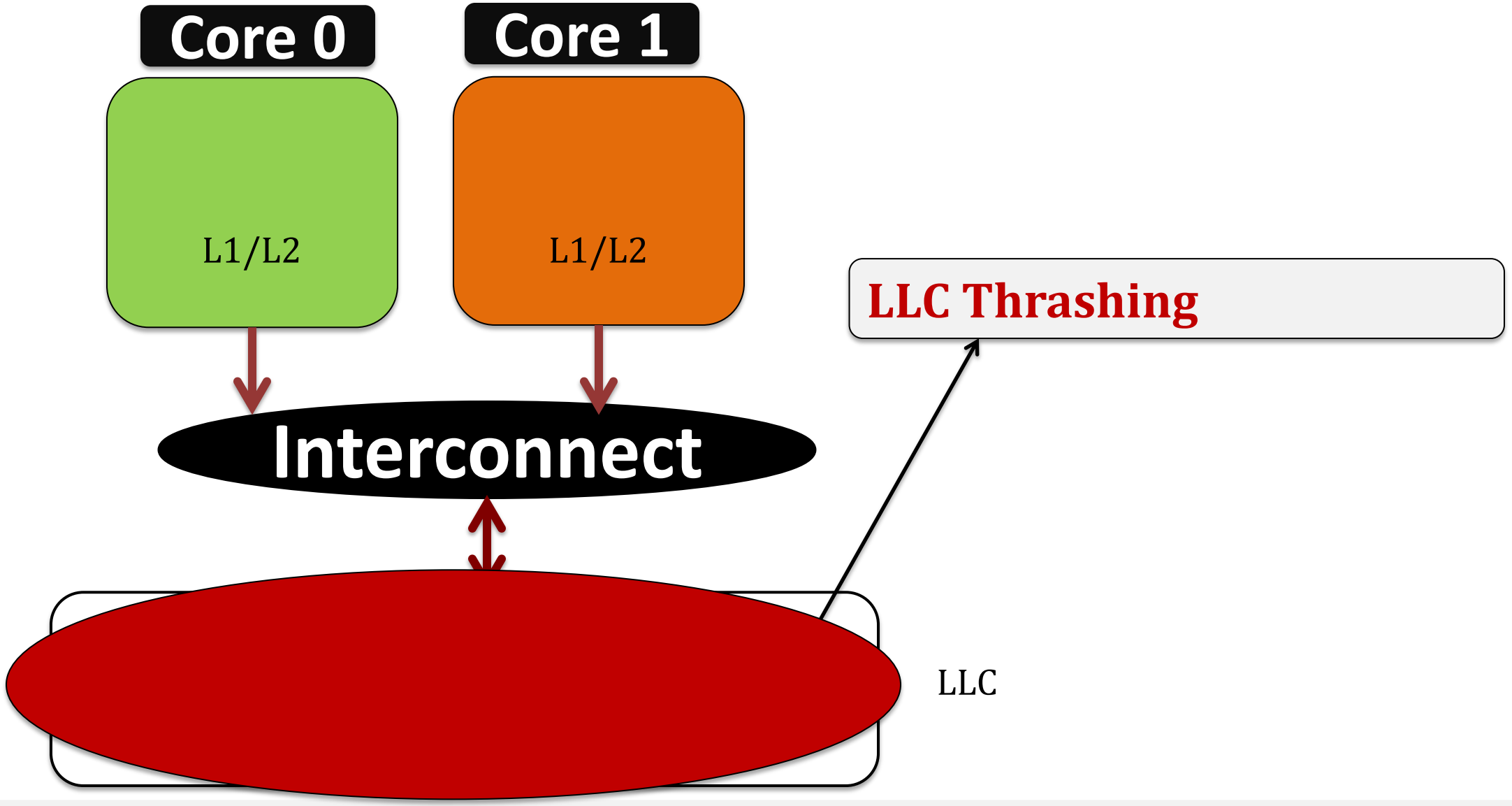
Shared Last-level Cache in Multi-core



Application Behavior @LLC



Application Behavior @LLC



Shared LLC

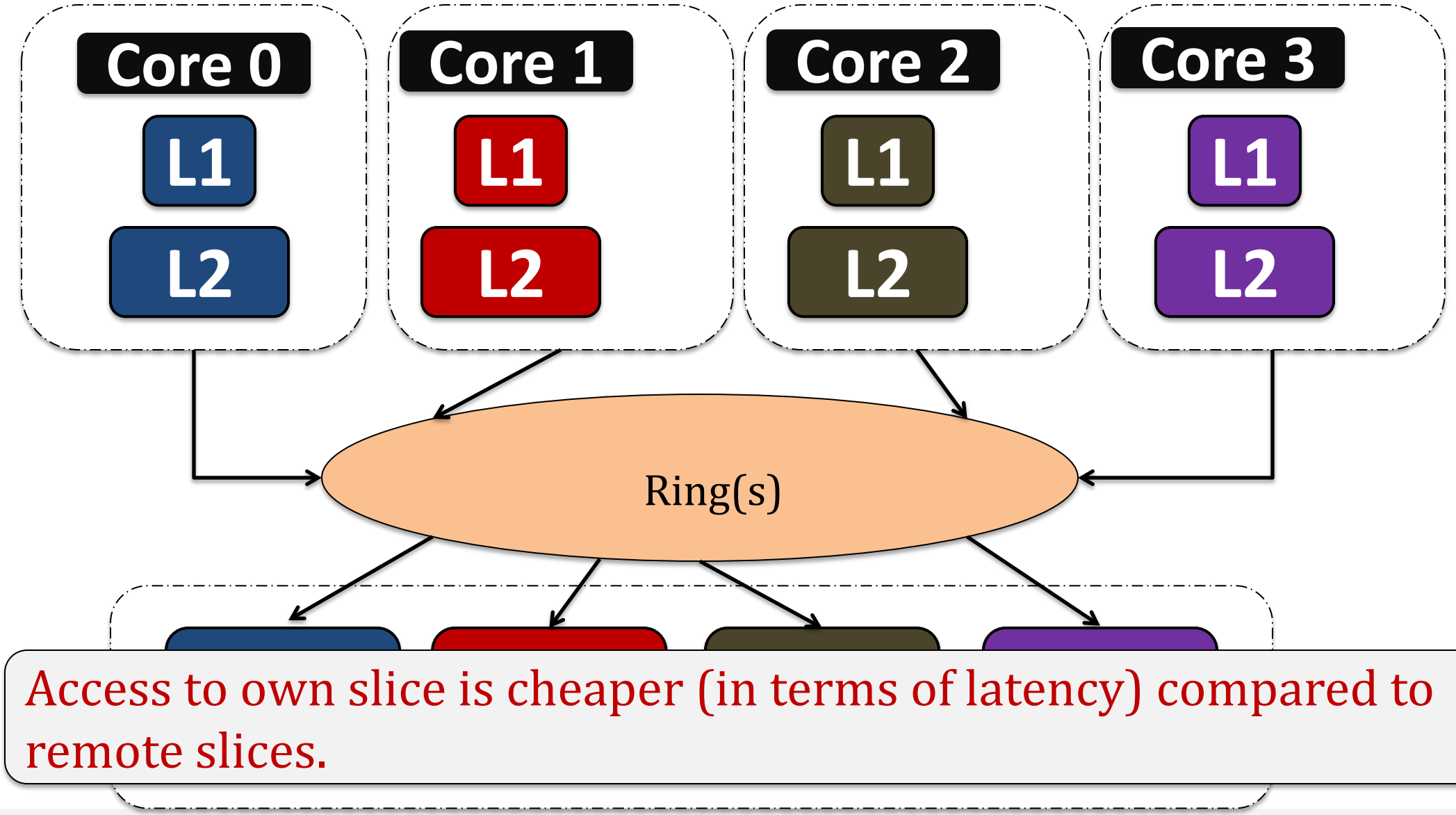
Shared LLC provides a good tradeoff for all kinds of apps.

Space unutilized by one app. can be utilized by other apps.

However bandwidth is an issue ☹

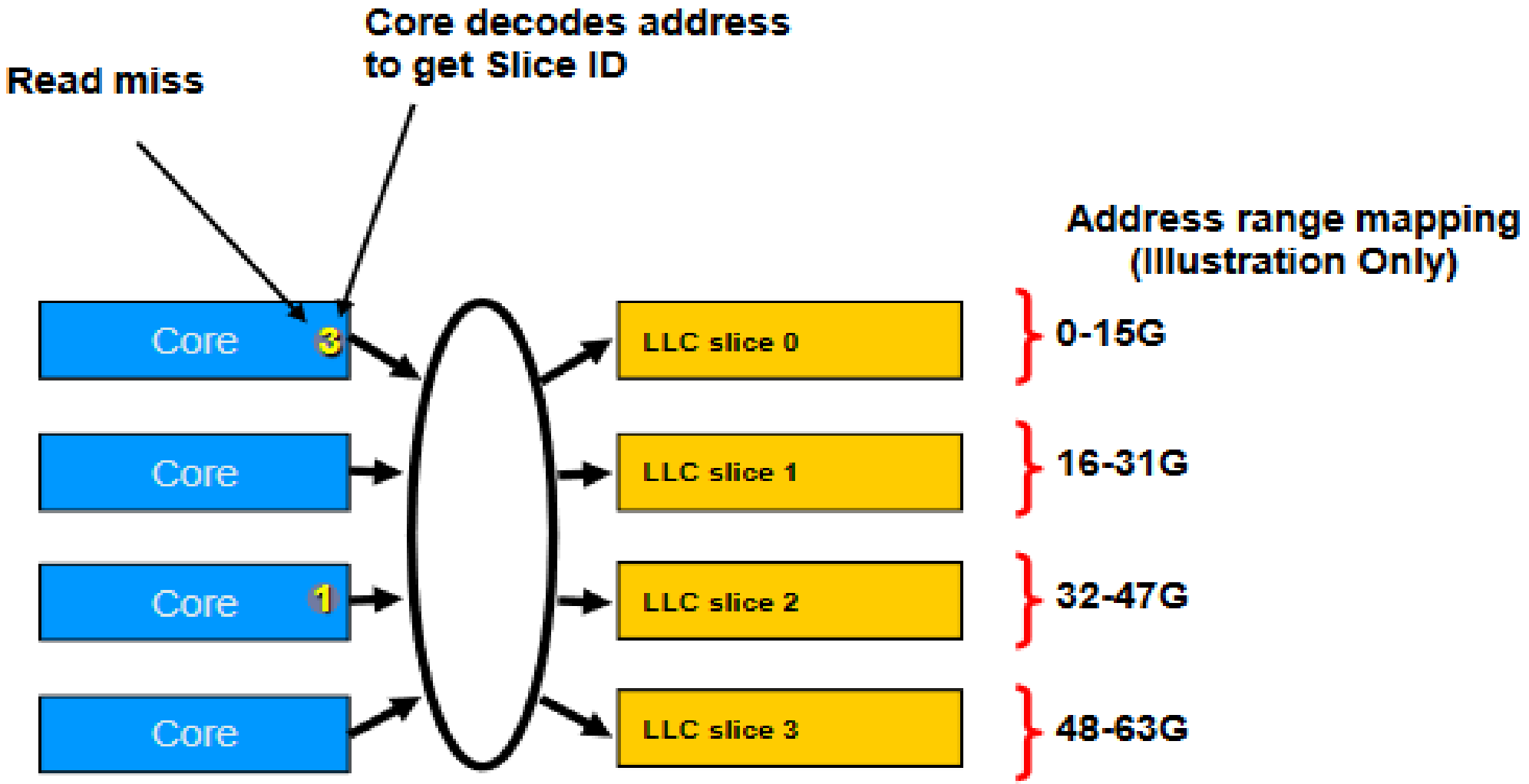
1000 monkeys: one banana

Sliced Shared LLC

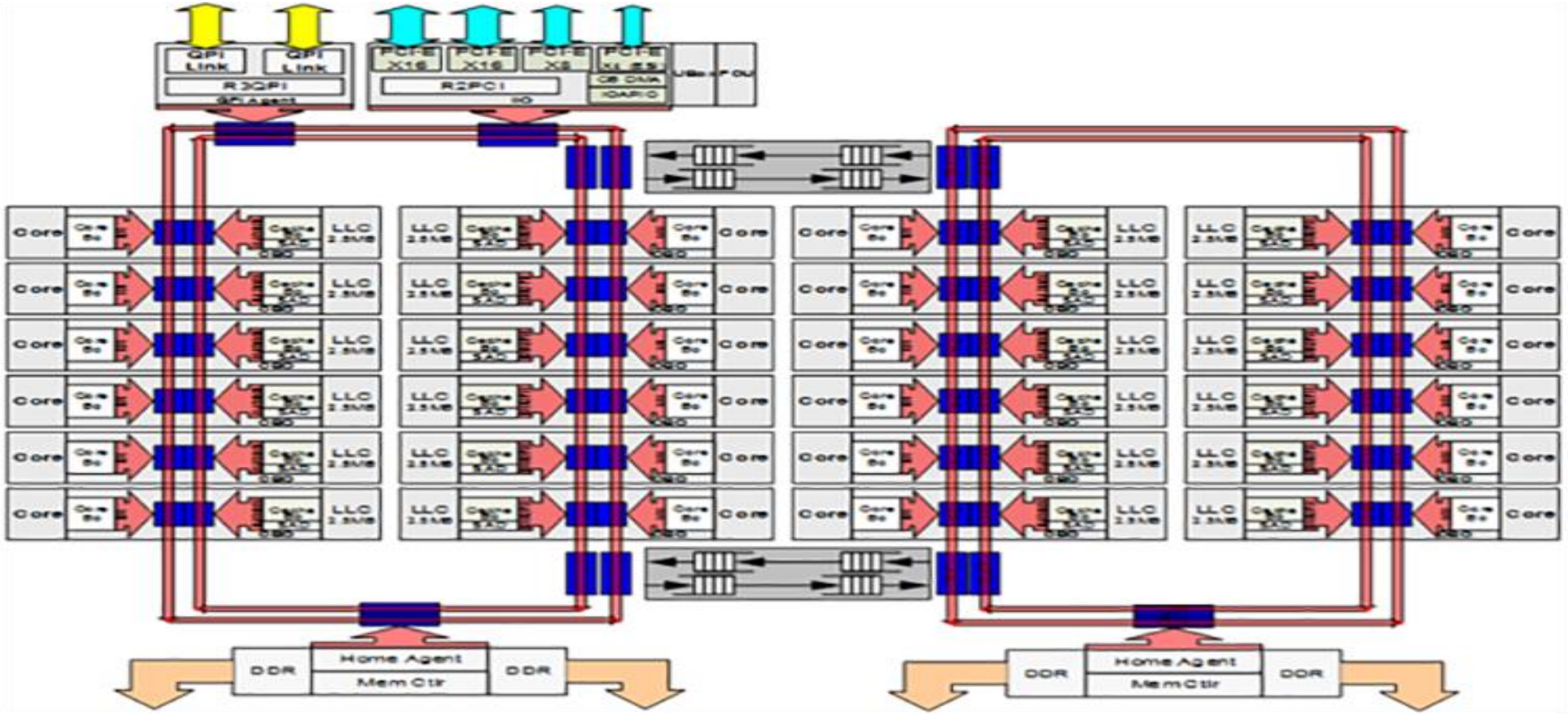


Access to own slice is cheaper (in terms of latency) compared to remote slices.

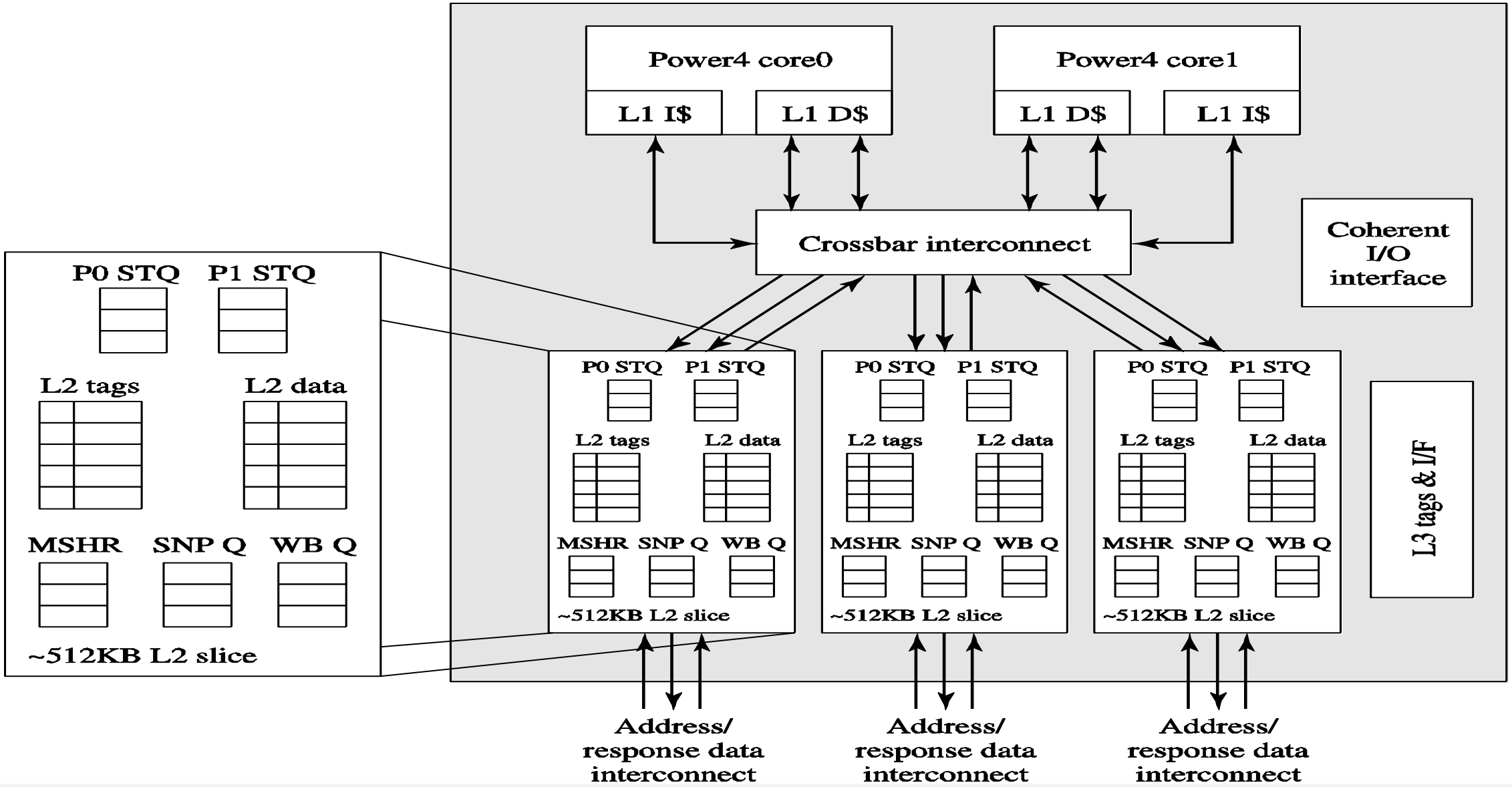
Intel's Sandybridge [Courtesy: Intel]



Intel Xeon [Courtesy: Intel]



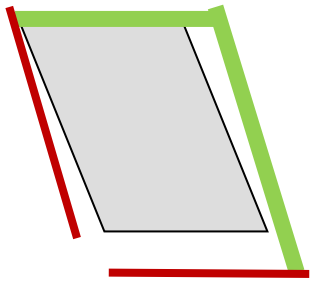
IBM POWER 4


Power4 core0
L1 I$
L1 D$
Power4 core1
L1 D$
L1 I$
Crossbar interconnect
Coherent I/O interface
P0 STQ
P1 STQ
L2 tags
L2 data
MSHR
SNP Q
WB Q
~512KB L2 slice
L3 tags & I/F
Address/ response data interconnect

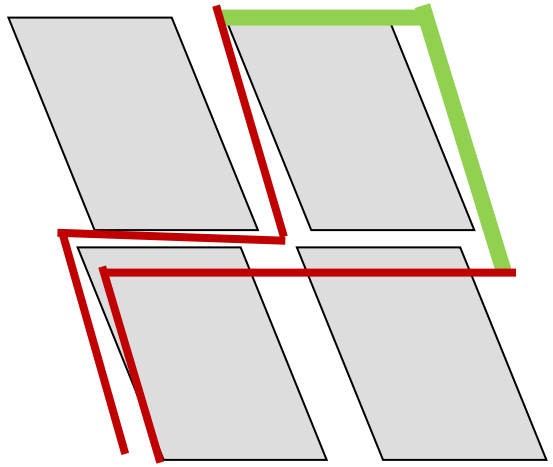
Read More about

Ring, bus, and crossbar

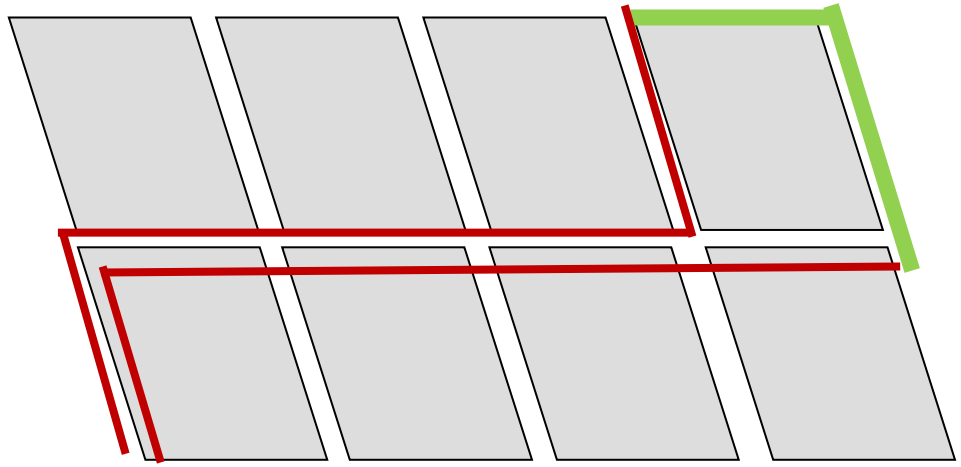
Non-uniform Cache Architecture [ASPLOS '02]



3 cycles, 1MB
180nm, 1999



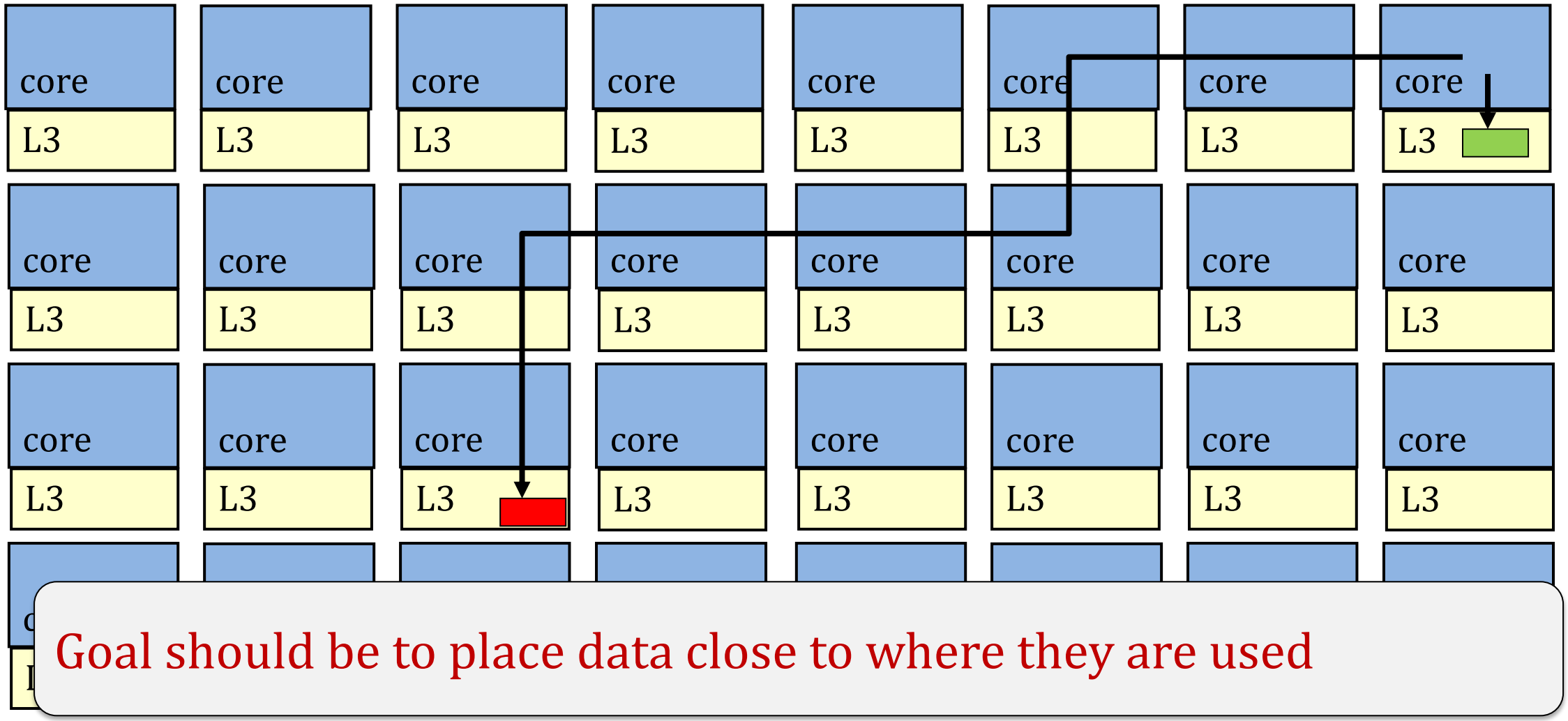
11 cycles, 4MB
90nm, 2004



24 cycles,16MB
50nm, 2010

Bigger and slower caches into smaller and faster slices.

Shared to Distributed [ISCA '09]



STATIC and DYNAMIC NUCA

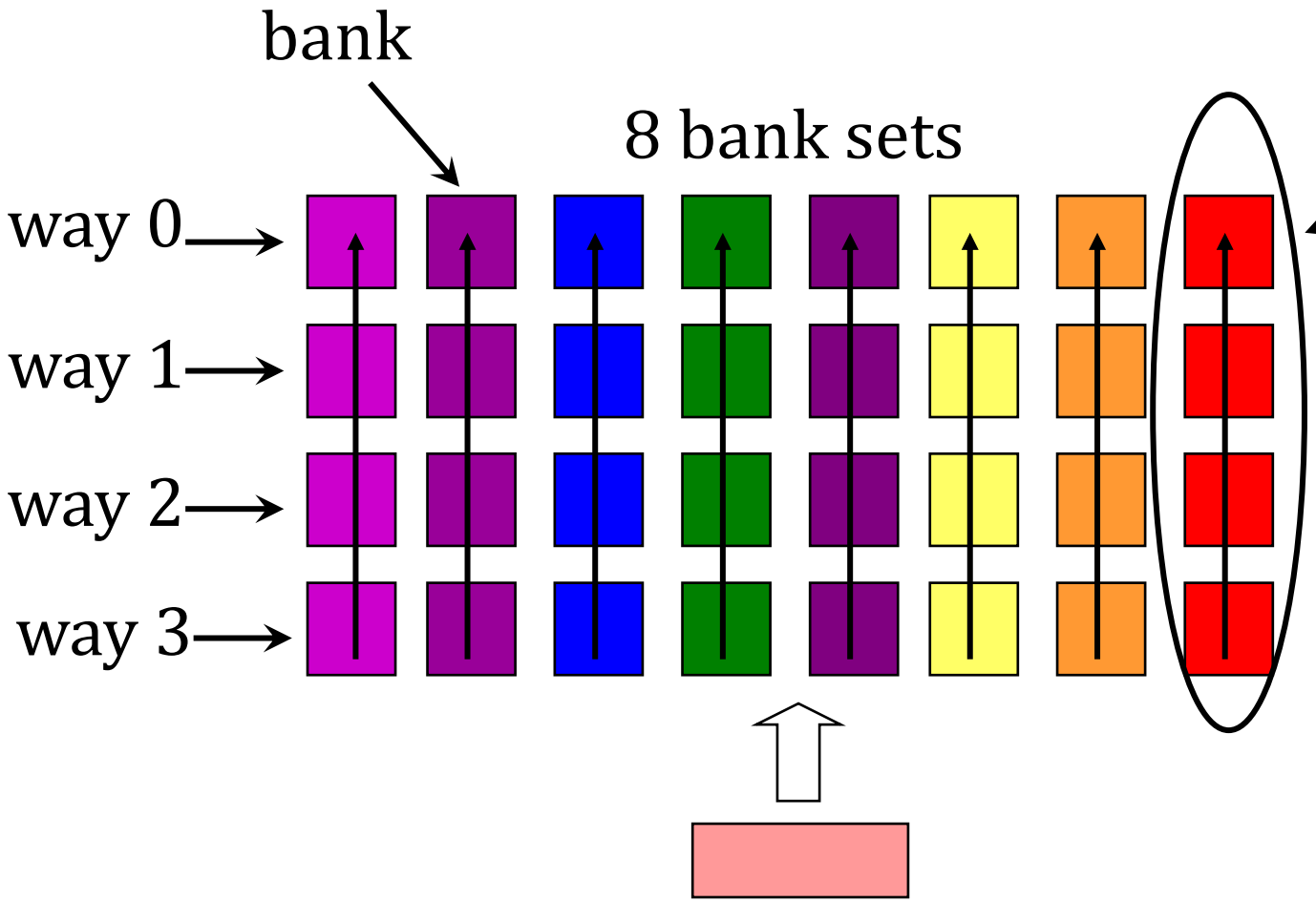
Static NUCA: Fixed mapping of cache lines into banks. Sets spread over banks.

Long wire latency to detect a cache hit.

Dynamic NUCA: Dynamic mapping of cache lines into banks. Ways spread over banks.

Blocks migrate from one bank to another. Move frequently used cache lines closer to CPU

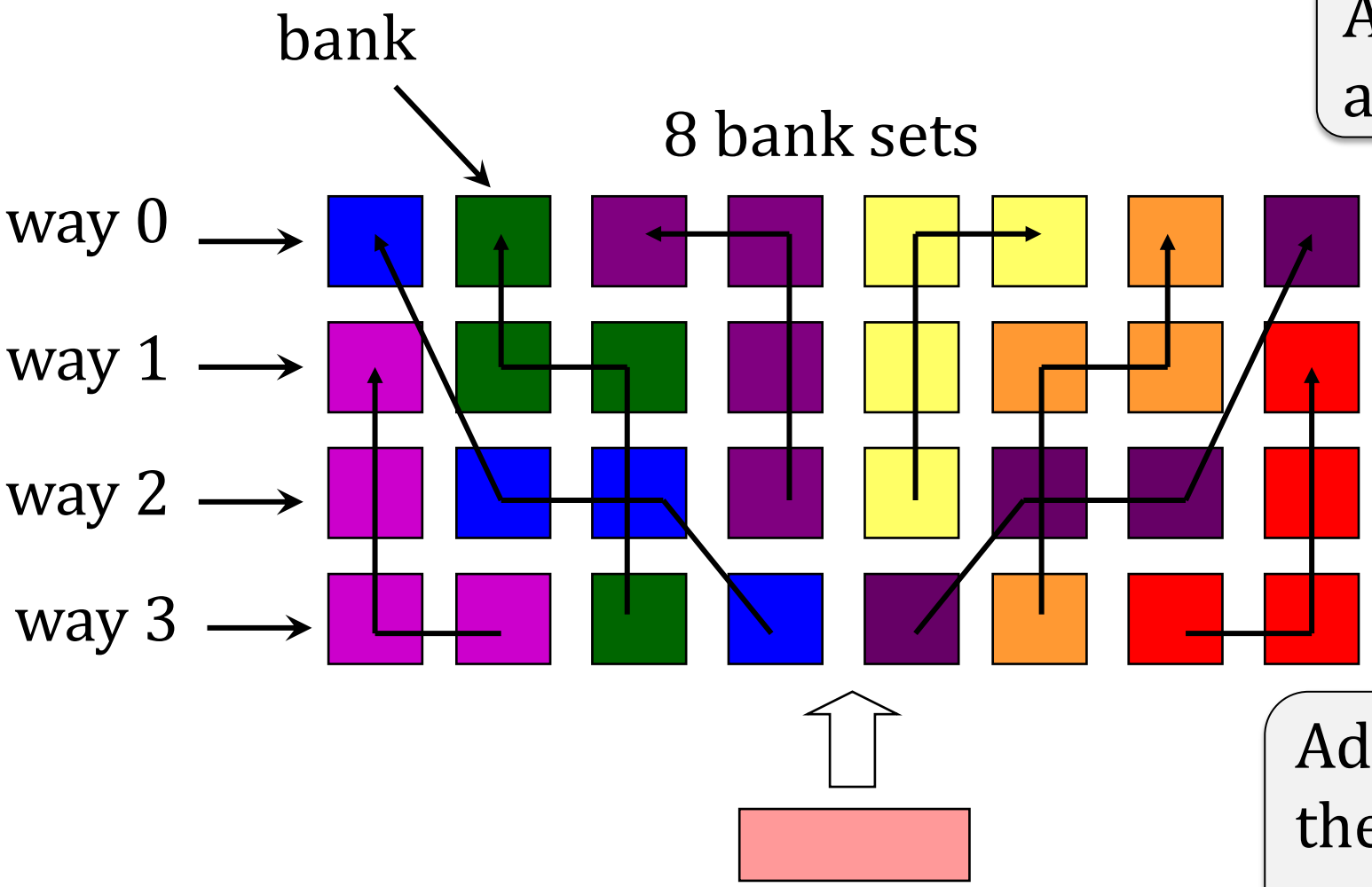
Simple Mapping – Bank sets to Banks



All 4 ways of each bank set need to be searched

Latencies to all bank sets are not same

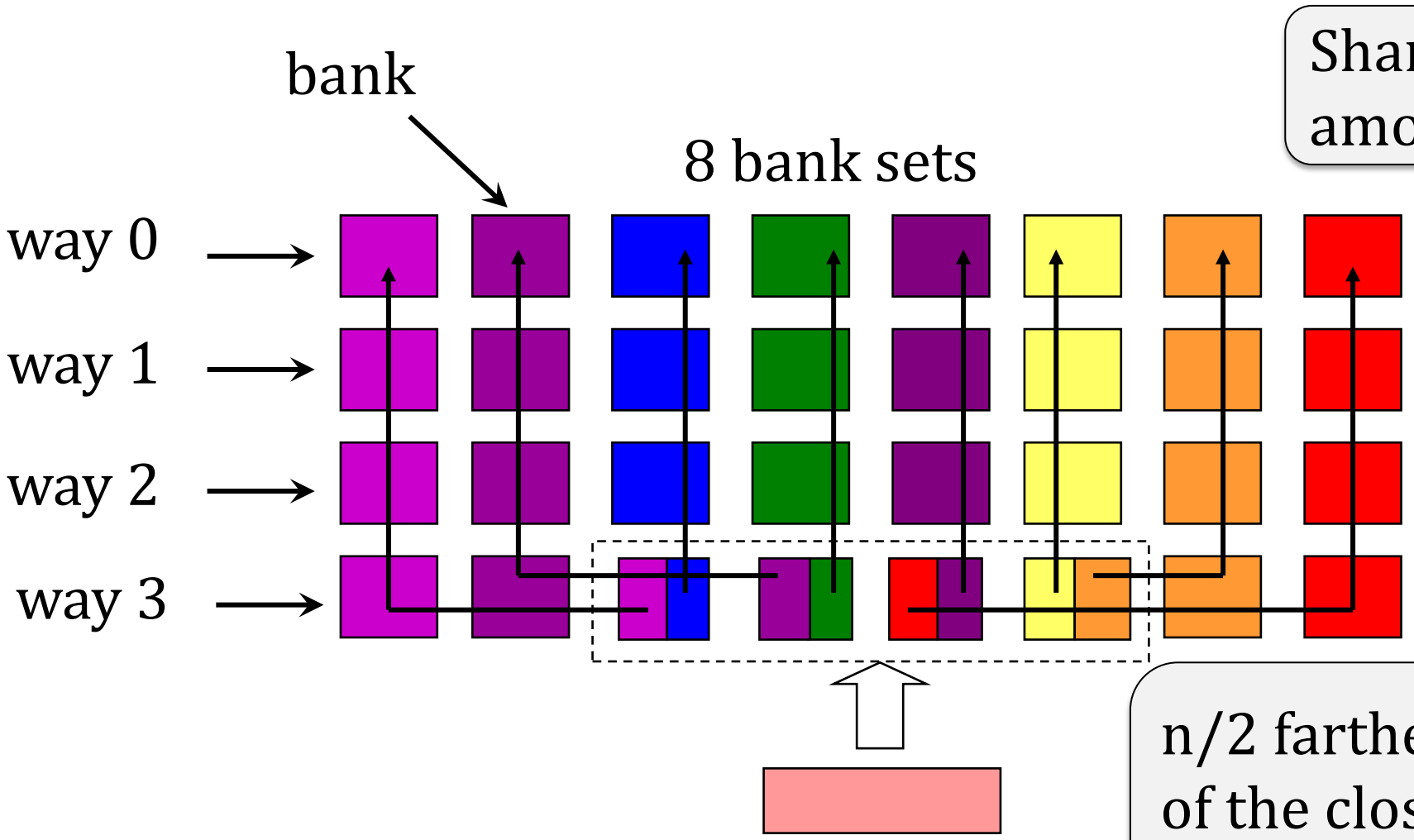
Fair Mapping



Average access time across all bank sets are equal

Additional hardware to store the mapping between banks and bank sets

Shared Mapping



Sharing the closest banks among multiple bank sets

n/2 farthest bank sets share half of the closest bank for one of the closest n/2 bank sets.

How to search a block?

Incremental Search: starting from closest bank till you find the block or miss.

Multicast search: Send requests to few or all banks of a bank set.

Limited search: Send requests to M banks and search in parallel and then incremental search.

What about Cache Access Time

$T(\text{Hit}) = T_{\text{search}} + T_{\text{bank\_hit}} + T_{\text{transer}}$

What about address mapping ? Instead of sets – bank sets and bank id

For many-core, T(hit) can become > DRAM-access