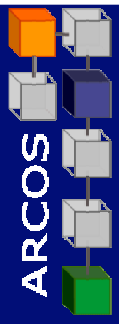


ARCOS

ersidad
s III de Madrid
uc3m.es

COMPUTER ARCHITECTURE

Cache Memory

Contents

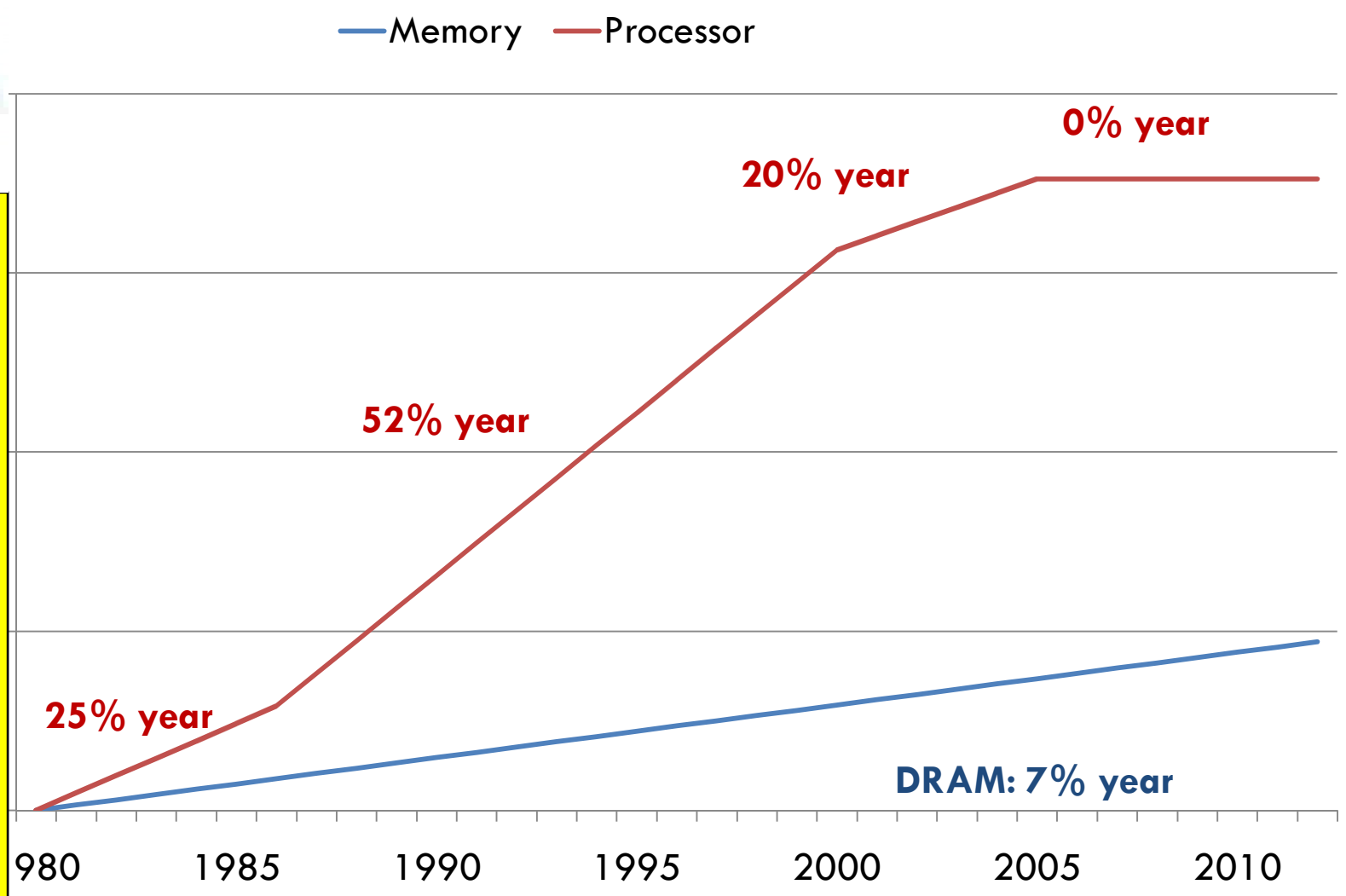
oduction.

he performance.

de-offs in cache design.

ic cache optimizations.

Performance evolution (1/latency)



Multi-core effect

Core i7

data access (64 bits) per cycle.

cores, 3.2 GHz → 25.6 x 10^9 access/sec

truction demand: 12.8 x 10^9 of 128 bits.

ak bandwidth: 409.6 GB/sec

AM Memory

DR2 (2003): 3.2 GB/sec – 8.5 GB/sec

DR3 (2007): 6.4 GB/sec – 17.06 GB/sec

DR4 (2014?): 17.05 GB/sec – 25.6 GB/sec

tions:

ulti-gate memory, pipelined caches, multi-level caches, per-core ches, instruction/data separation.

ality principle:

- rogram property exploited in hardware design.
- rograms access to a relatively small portion of ddress space.

es of locality:

- **emporal locality**: Recently accessed elements tend to e accessed again.
  - Examples: loops, variable reuse, …
- **patial locality**: Elements next to a recently accessed ement ten to be accessed in near future.
  - Examples: sequential instruction execution, arrays, …

Situation (2008)

AM – Static RAM

ccess time: 0.5 ns – 2.5 ns

ost per GB: 2000$ - 5000$

AM – Dynamic RAM

ccess time: 50ns – 70 ns

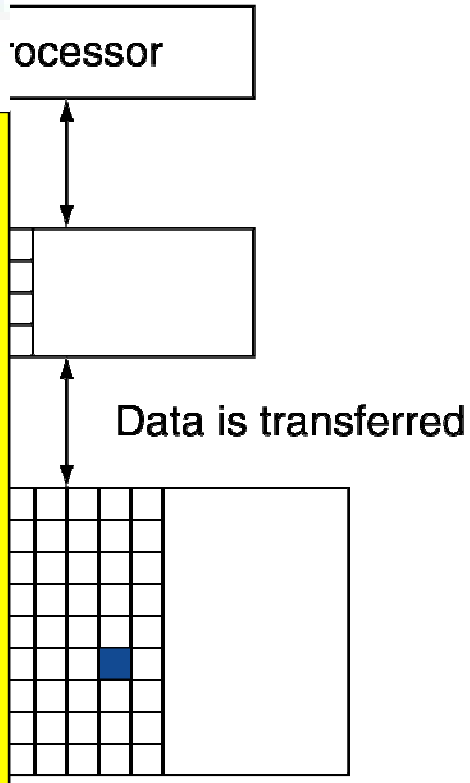
ost per GB: 20$ - 75$

gnetic disk

ccess time: 5,000,000 ns – 20,000,000 ns

ost per GB: 0.20 $ - 2$

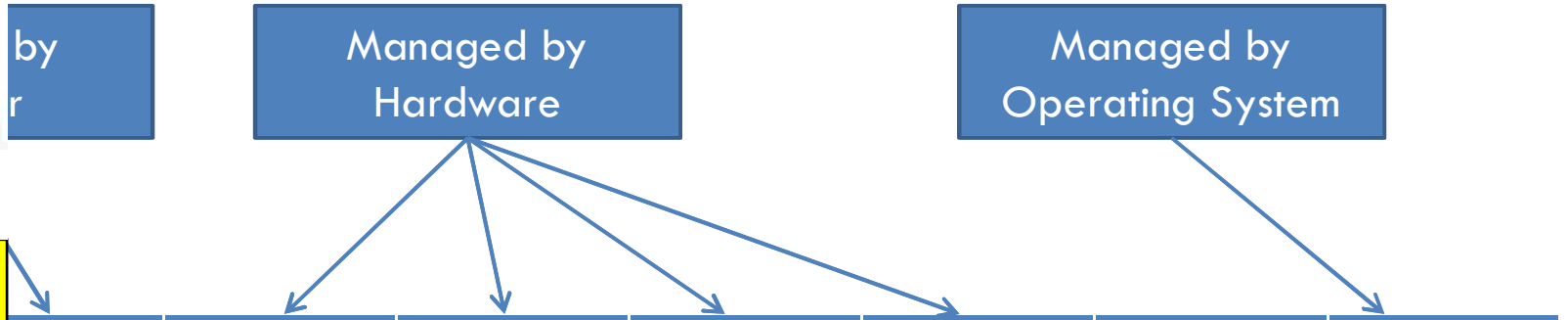
Memory hierarchy



- **Block or line**: Unit of copy.
  - Usually several words.
- If accessed data present in higher level:
  - **Hit**: Delivered by higher level.
    - Hit rate = Hits / accesses.
- If accessed data is missing.
  - **Miss**: Block copied from lower level.
    - Needed time → Miss penalty.
    - Miss rate = Misses / accesses = 1 – Hit rate

Cache interconnection

Example: Haswell Core i7



gisters	L1I	L1D	L2	L3	DRAM	Disk
	32KB	32KB	4x256KB	8MB	32GB	1.8TB
le	4-5 cycles	4-5 cycles	12 cycles	36 cycles	36 cycles + 57 ns	2.8 ms

Goal: Give the illusion of large, fast, and cheap memory

Allow programs to address space scalable to disk size at register file speed.

Contents

oduction.

he performance.

de-offs in cache design.

ic cache optimizations.

rage memory access time:

$= t_H + (1-h) \cdot t_M$

s penalty:

me to replace a block and deliver to CPU.

ccess time:

Time to get from lower level.

Dependent on lower level latency.

ansfer time:

Time to transfer a block.

Depending on bandwidth across levels.

execution time

PU cycles + Memory stall cycles) x Cycle time

cycles

x CPI

Beware: May be different for reads and writes.

ory stall cycles

sses number x Miss penalty

x Misses per instruction x Miss penalty

x Instruction Memory access x Miss rate x Miss penalty

re:

→ Instruction Count.

I → Cycles per instruction.

Contents

oduction.

he performance.

de-offs in cache design.

ic cache optimizations.

Four questions on memory hierarchy

Where can a block be placed in the upper
l?

ock placement.

How is a block found in the upper level?

ock identification.

Which block should be replaced on a miss?

ock replacement.

What happens on a write?

/rite strategy.

Q1: Block Placement

t mapping.

acement → block MOD blocks_in_cache

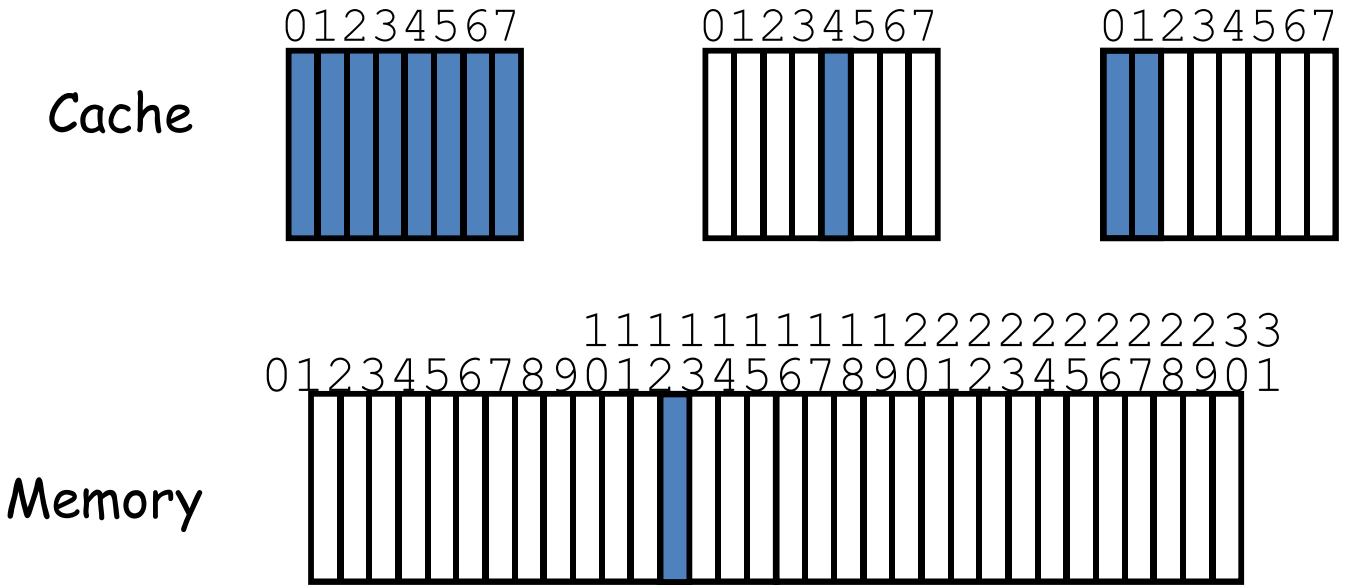
associative mapping.

acement → Anywhere.

ssociative mapping.

t placement → block MOD number_of_sets.

acement within set → Anywhere.



Q2: Block Identification

k Address:

ıg: Identifies the entry address.

Validity bit in every entry to flag if content is valid.

dex: Selects address.

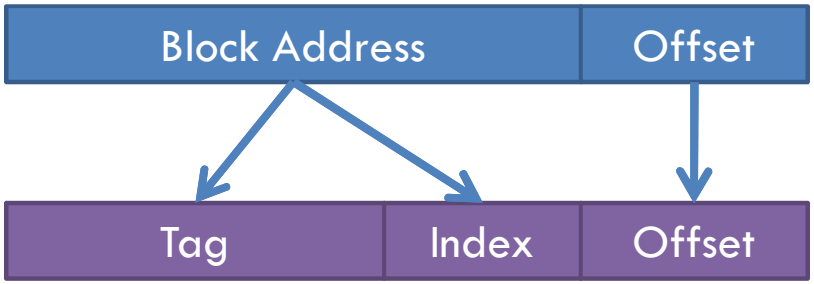
k offset:

elect data within a block.

ner associativity means:

ss bits for Index.

ore bits for Tag.



Q3: Block Replacement

vant for associative and set associative mappings:

ndom.

Easy to implement.

RU: Less Recently Used.

Increasing complexity as associativity increases.

FO: First In First Out.

Approximates LRU with lower complexity.

Misses per1000 instr., SPEC 2000

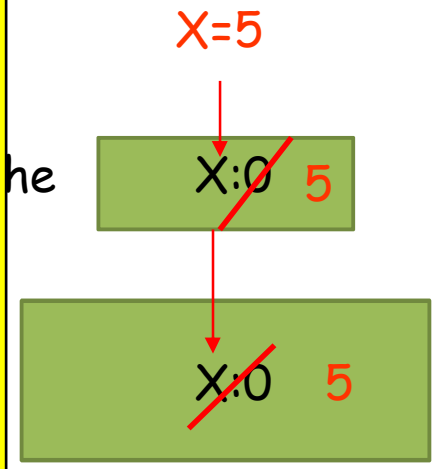
2 ways			4 ways			8 ways		
U	Rand	FIFO	LRU	Rand	FIFO	LRU	Rand	FIFO
4.1	117.3	115.5	111.7	115.1	113.3	109.0	111.8	110.4
03.4	104.3	103.9	102.4	102.3	103.1	99.7	100.5	100.3
92.2	92.1	92.5	92.1	92.1	92.5	92.1	92.1	92.5

From Hennessy & Patterson, 5Ed Appendix C

Q4: Write strategy

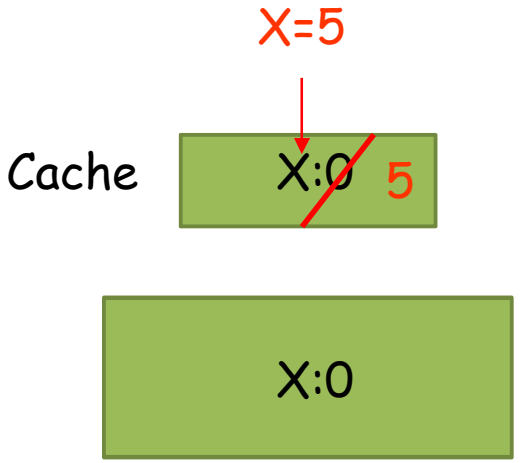
ough

rites sent to bus and
ory.
to implement.
rmance issues in SMPs .



Write back

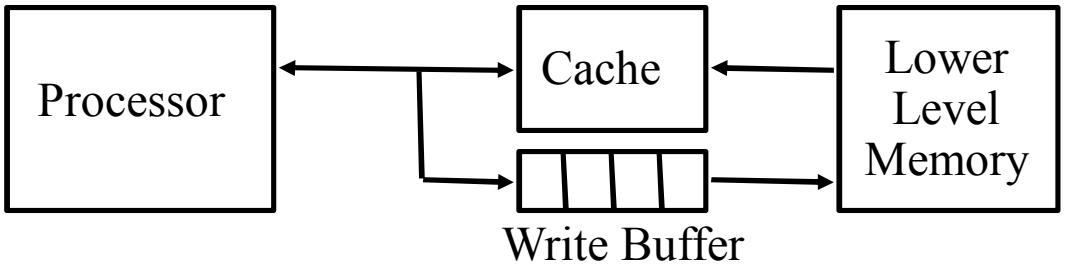
- Many writes are a hit.
- Write hits **not** sent to bus and memory.
- Propagation and serialization issues.
- More complex.



Q4: Write strategy

	Write-Through	Write-Back
olicy	Data written to cache blocks. Also written to next level in memory.	Data only written to cache. Update next levels when block evicted from cache
ugging	Easy	Difficult
ite on iss?	No	Yes
peated s sent to t level?	Yes	No

Write Buffers in caches with write through



y a buffer?

avoid CPU stalls.

y a buffer instead
register?

Write bursts are
mmon.

Are RAW hazards possible?

- Yes.
- Alternatives:
  - Flush buffer before a read.
  - Check buffer before a read.

Example: Opteron

Miss penalty and Out-of-Order execution

s penalty definition:

iss total latency.

xposed latency (generating CPU stall).

s penalty:

emory stalls / Instructions

Misses/Instructions) x (Total latency – overlapped
tency)

Contents

oduction.

he performance.

de-offs in cache design.

ic cache optimizations.

Cache basic optimizations

uce miss rate.

- arger block size.
- arger cache size.
- igher associativity.

uce miss penalty.

- ulti-level caches.
- ioritize read over writes.

uce hit time.

- void address translation in cache indexing.

1: Larger block size

l: Reduce miss rate.

prove spatial locality exploitation.

eases miss penalty.

pon a miss, larger blocks need to be transferred.

ore misses due to cache with less blocks.

nce needed:

gh latency and high bandwidth memory:

Increase block size.

w latency and low bandwidth memory:

Reduced block size.

Miss rate and block size

2: Larger cache size

l: Reduce miss rate.

ore data fit in cache.

y increase hit time.

ore time needed to find block.

her cost.

her energy consumption.

d to find balance:

pecially in on-chip caches.

3: Higher associativity

al: Reduce miss rate.

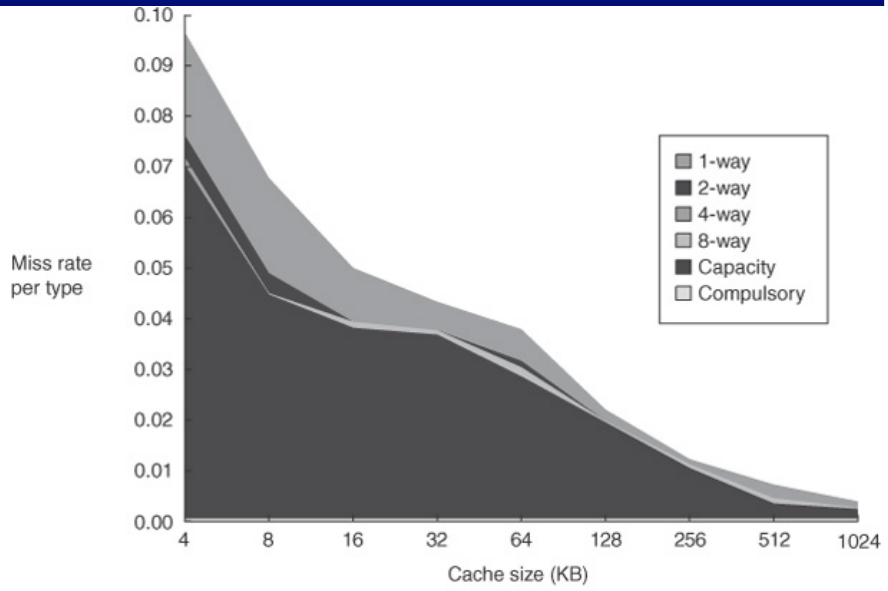
ess conflicts as more ways a set can be used.

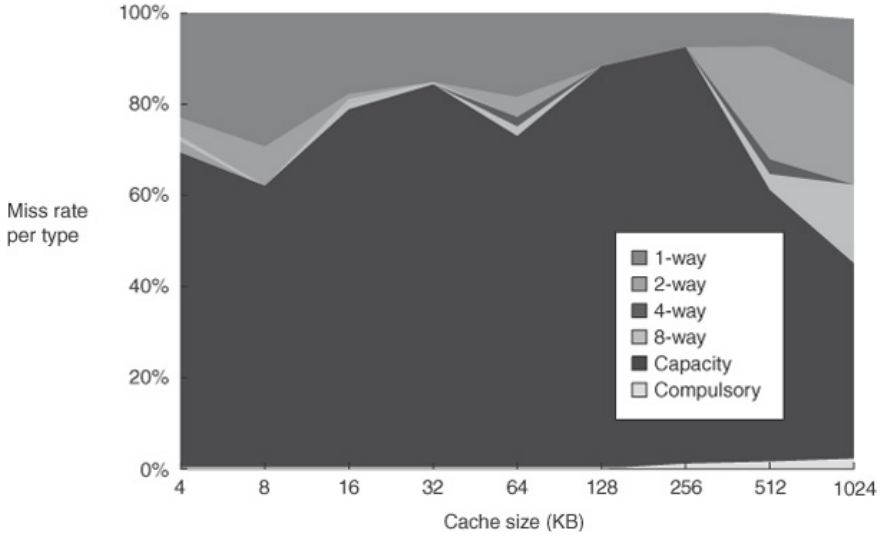
y increase hit time.

ore time needed to find block.

sequence:

ways ≈ Fully associative





l: Reduce miss penalty.

lution:

creasing performance gap between DRAM and CPU.

iss penalty cost increased over time.

rnatives:

ster caches.

rger caches.

tion:

th of them!

veral cache levels.

Local and global miss rate

rage access time:

t time$_{L1}$ + Miss rate$_{L1}$ x Miss penalty$_{L1}$

t time$_{L1}$ + Miss rate$_{L1}$ x
it time$_{L2}$ + Miss rate$_{L2}$ x Miss penalty$_{L2}$)

l miss rate:

isses at a cache level over accesses to that cache level.

: Miss rate$_{L1}$

: Miss rate$_{L2}$

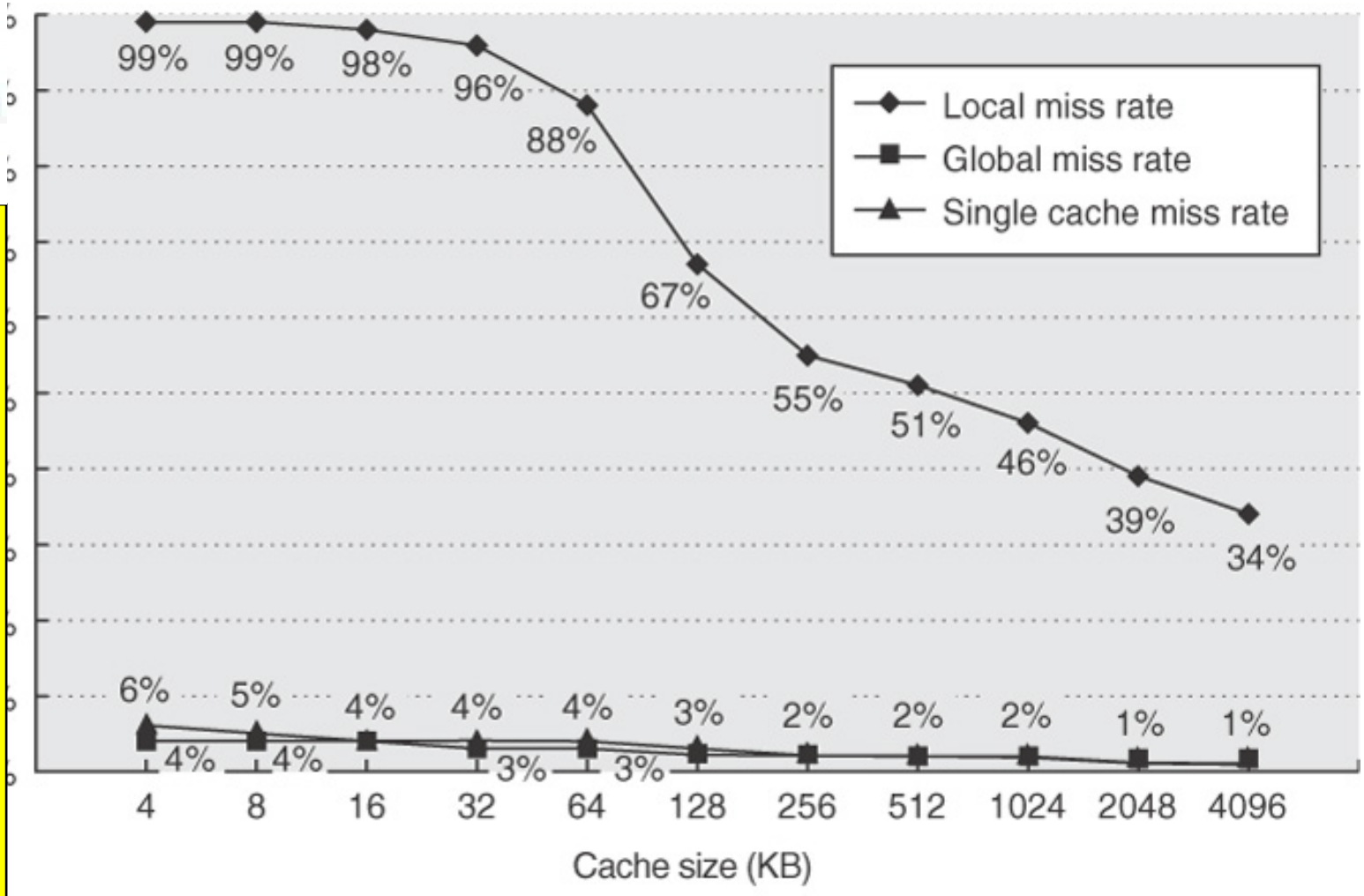
bal miss rate:

isses at a cache level over **all** memory accesses.

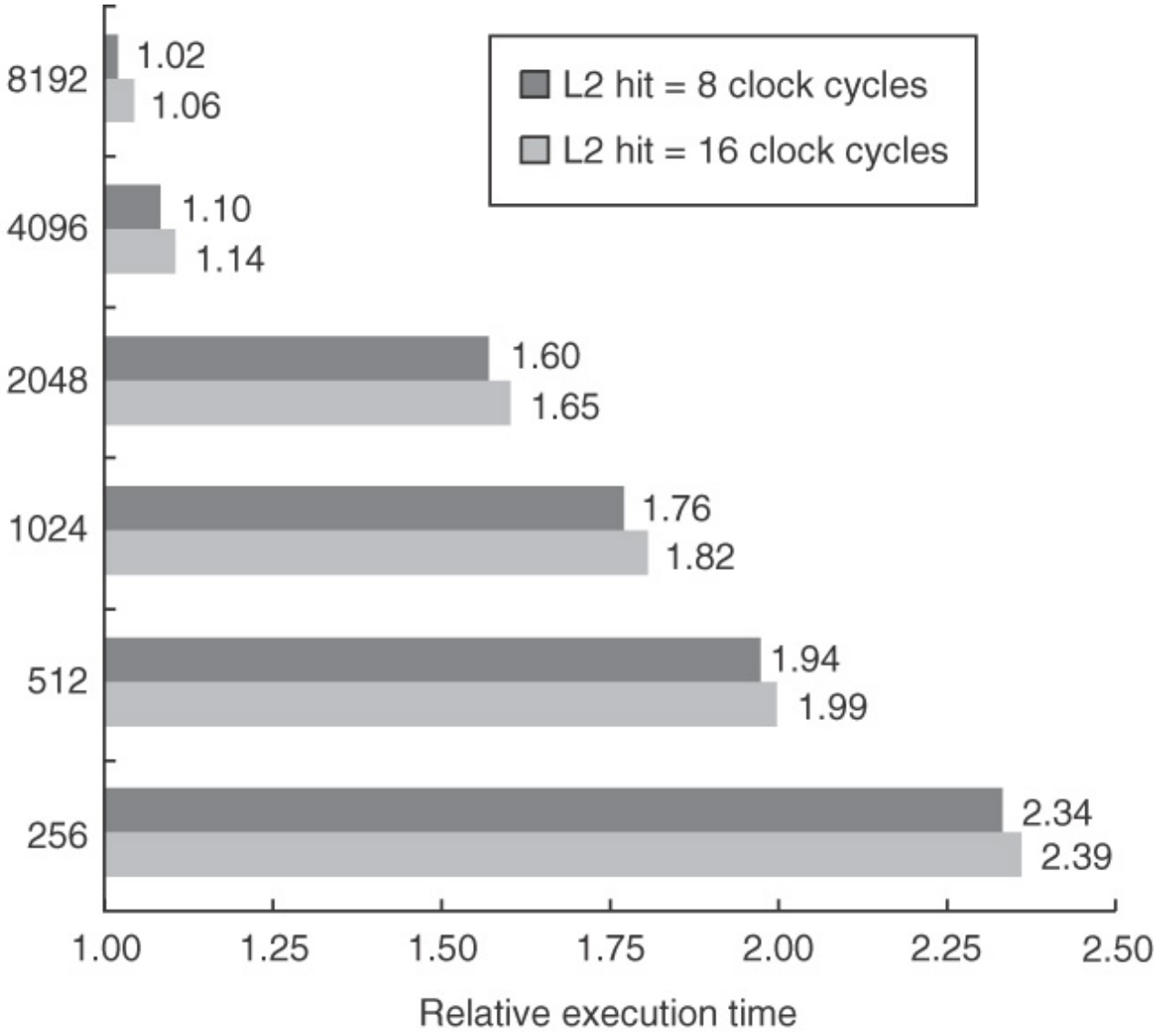
: Miss rate$_{L1}$

: Miss rate$_{L1}$ x Miss rate$_{L2}$

Miss rate versus cache size for multilevel caches



Relative execution time by second-level cache size



On-chip multilevel caches

Nehalem 4-core processor

ore: 32KB L1 I-cache, 32KB L1 D-cache, 512KB L2 cache

5: Prioritize read misses over writes

l: Reduce miss penalties.

oid that a read miss has to wait until writes are completed.

e-through caches:

rite buffer could contain an updated value for the read ddress.

A) Wait until write buffer is empty.

B) Check contents of write buffer.

- Continue with read miss if no conflict with buffer.

e-back caches:

read miss could replace a modified block.

Copy modified block to buffer, read, and dump block to memory.

Apply options A or B to buffer.

6: Avoid Address Translation during Indexing

l: Reduce hit time.

nslation process:

rtual address → Physical address.

ay require additional accesses to memory.

Or at least to TLB.

: Optimize the most common case (hits).

se virtual addresses for the cache.

ks:

dexing the cache.

mparing tags.

Problems with virtual caches (I)

tection:

age-level protection checked during virtual-to-physical anslation.

olution: Copy protection info from TLB o misses.

cess switch.

irtual addresses refer to different physical addresses.

Old process virtual addresses.

olutions:

Flush the cache.

Add to cache address a PID tag.

Miss rate in virtual caches

Problems with virtual caches (II)

sing:

vo different virtual addresses for the same physical
ddress.

nti-aliasing hardware: to guarantee that every cache
ock corresponds to a unique physical address.

Check multiple addresses and invalidate.

ge coloring: Force all aliases to have identical their n last
ts.

Makes impossible two alias to be at the same time in cache.

addresses:

O typically uses physical addresses.

apping to virtual addresses to interact with virtual caches.

ution:

irtual indexing and physically tagging.

ks:

dexing the cache → Use page offset.

This part is identical in physical and virtual addresses

omparing tags → Use translated physical address.

Tag matching uses phyical address.

hing as a **solution** to mitigate the memory wall.

he performance due to **locality principle** (spatial temporal).

s penalty dependent on access time and transfer .

key dimensions in **cache design**:

ock placement, block identification, block replacement and rite strategy.

basic cache **optimizations**:

educe miss rate: larger block size, larger cache size, gher associativity.

educe miss penalty: multilevel caches, prioritize reads ver writes.

educe hit time: Avoid address translation when indexing.

References

nputer Architecture. A Quantitative Approach.
n Edition.

nessy y Patterson.

tions: B.1, B.2, B.3

rcises:

B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9, B.10, B.11