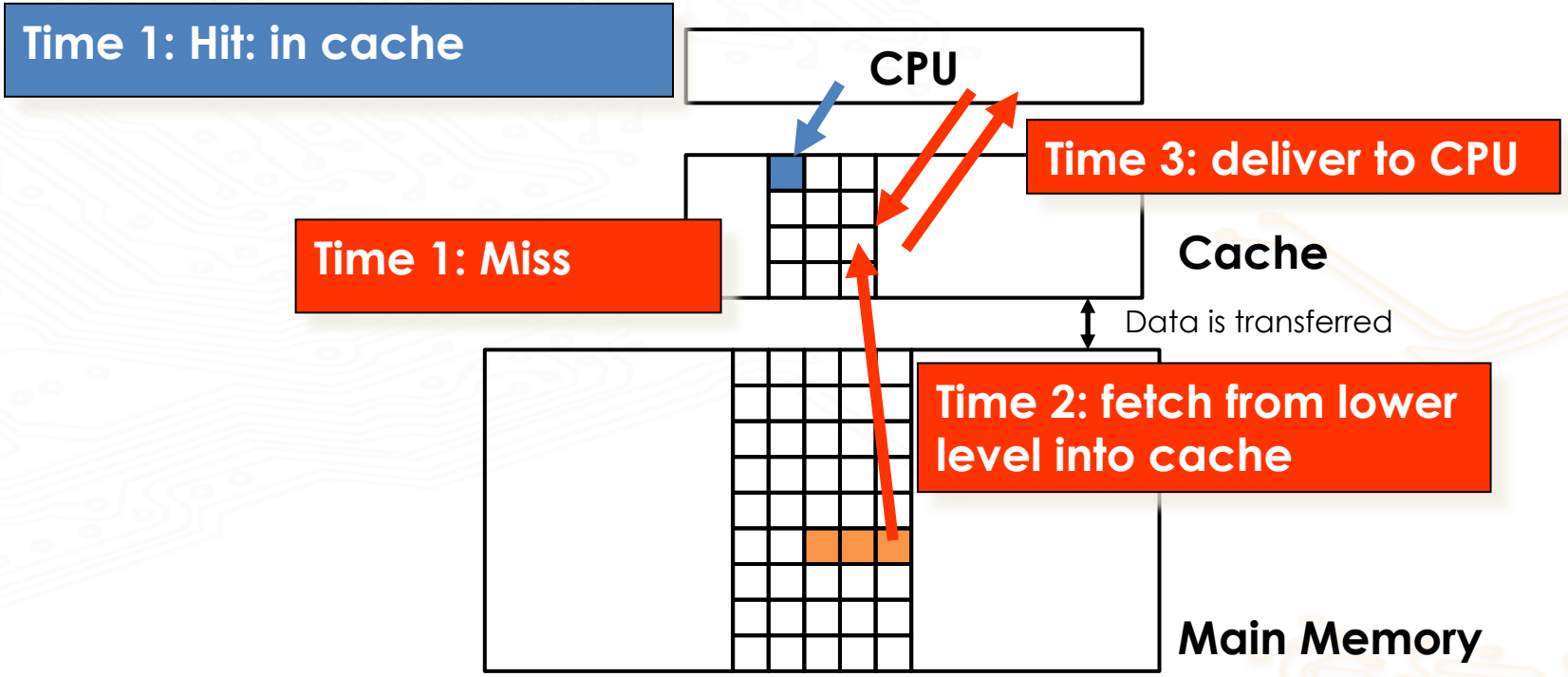
CE/CZ 3001:
Advanced Computer Architecture

Module 5: Memory Systems

- cache performance

Asst Prof Liu Weichen
School of Computer Science and Engineering
Nanyang Technological University, Singapore

Cache Example



Hit time = Time 1

Miss penalty = Time 2 + Time 3

Cache and its performance (Part 1/2)

- Caches
  - cache hit: life is made easy and improves performance
  - cache miss: fetch from next level (Apply recursively if multiple levels)
- What is performance impact?
  - Miss penalty leads to reduction of performance

Cache and its performance (Part 2/2)

- Miss penalty
  - Detect miss: 1 or more cycles
  - Find victim (the line to be replaced): 1 or more cycles
    - Write back if dirty
  - Request line from next level: several cycles
  - Transfer line from next level: several cycles
    - (block size) / (bus width)
  - Fill line into data array, update tag array: 1+ cycles
  - Resume execution
- In practice: 6 cycles to 100s of cycles

Cache Miss (Part 1/2)

- Cache miss rate is determined by:
  - Temporal locality
  - Spatial locality
  - Cache organization
    - Block size, associativity, number of sets

Cache Miss (Part 2/2)

- Reasons for cache miss
  - Compulsory miss
    - First-ever reference to a given block of memory
  - Capacity miss
    - Working set exceeds cache capacity
    - Useful blocks (with future references) displaced
  - Conflict miss
    - Placement restrictions (not fully-assoc.) cause useful blocks to be displaced
    - Think of as *capacity within set*

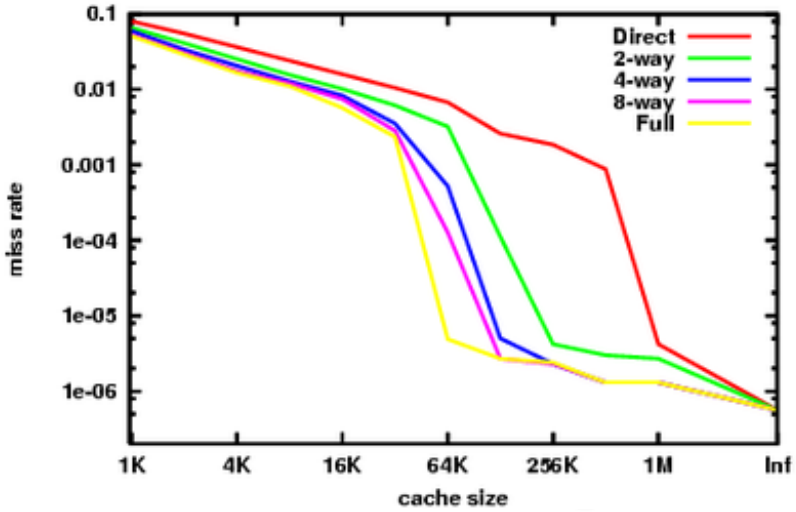
Cache Miss Rate Effects (Part 1/2)

- Number of blocks (sets x associativity)
  - Bigger is better: fewer conflicts, greater capacity
- Associativity
  - Higher associativity reduces conflicts
  - Very little benefit beyond 8-way set-associative
- Block size
  - Larger blocks exploit spatial locality
  - Usually: miss rates improve until 64B-256B
  - 512B or more miss rates get worse
    - Fewer placement choices: more conflict misses
    - High miss penalty

Cache Miss Rate Effects (Part 2/2)

- Subtle tradeoffs between cache organization parameters
  - Large blocks reduce compulsory misses but increase miss penalty
  - Large blocks increase conflict misses
  - Associativity reduces conflict misses
  - Associativity increases access time

Cache Miss and Performance (Part 1/2)



SPEC CPU2000 benchmarks, as collected by Hill and Cantin

"Cache performance of SPEC CPU2000". Cs.wisc.edu. Retrieved 2010-05-02.

Beyond 8-way set-associative is not generally done

Cache Miss and Performance (Part 2/2)

- How does this affect performance?
- Execution Time

$$= \underbrace{\text{Instruction count}}_{\text{(IC)}} \times \underbrace{\frac{\text{Cycles}}{\text{Instruction}}}_{\text{(CPI)}} \times \underbrace{\frac{\text{Time}}{\text{Cycle}}}_{\text{(cycle time)}}$$

- Assuming cache hit costs are included as part of the normal CPU execution cycle, then

$$\text{CPU time} = \text{IC} \times \text{CPI} \times \text{cycle time}$$

$$= \text{IC} \times \underbrace{(\text{CPI}_{\text{ideal}} + \text{Memory-stall cycles})}_{\text{CPI}_{\text{stall}}} \times \text{cycle time}$$

- Memory-stall cycles = memory accesses/program × miss rate × miss penalty

Impacts of cache performance

- A processor with a CPI_{ideal} of 2, a 100 cycle miss penalty,
- 36% load/store instr's, and 2% IM and 4% DM miss rates

Memory-stall cycles = Memory stall cycles in IM + memory stall cycles in DM

Memory stall cycles in IM =2% (miss rate of IM)× 100 (Miss penalty) =2
(All instructions has to be fetched from IM)

Memory stall cycles in DM =
36% (LW/SW) × 4% (miss rate)× 100 (miss penalty) = 1.44

So CPI_{stalls} = 2 + 2+ 1.44 = **5.44** (more than twice the CPI_{ideal} !)

- What if the CPI_{ideal} is reduced to 1?

For ideal CPI = 1, then CPI_{stall} = 4.44 and the amount of execution time spent on memory stalls would have risen from 3.44/5.44 = 63% to 3.44/4.44 = 77%

Example 2: Multiple levels of cache (Part 1/2)

- First level cache :-L1 caches
- *and a* second level of caches – normally a unified L2 cache (i.e., it holds both instructions and data)
- and in some cases even a unified L3 cache (LLC - last level cache)

Example 2: Multiple levels of cache (Part 2/2)

- For our example, CPI_{ideal} of 2,
  - 100 cycle miss penalty (to main memory)
  - 25 cycle miss penalty (to Unified L2 cache)
  - 36% load/stores,
  - 2% miss-rate for L1 I-Mem and 4% miss-rate for L1 D-Mem
  - 0.5% Unified L2 cache miss rate.

Note: every instruction will access the instruction memory for instruction fetch (100%)
only load/store instructions will access the data memory (36%)

$$CPI = CPI_{ideal} + CPI_{L1Inst\_miss} + CPI_{L1data\_miss} + CPI_{L2Inst\_miss} + CPI_{L2data\_miss}$$
$$= CPI_{ideal} + CPI_{L2\_hit\_inst} + CPI_{L2\_hit\_data} + CPI_{MM\_inst} + CPI_{MM\_data}$$
$$= 2 + 2\%*25 + 36\%*4\%*25 + 2\%*0.5\%*100 + 36\%*4\%*0.5\%*100$$
$$= 2+0.5+0.36+0.01+0.0072$$
$$= 2.8772 \text{ cycles/inst. (as compared to 5.44 (no L2cache))}$$

Average memory access time (AMAT) (Part 1/2)

- Average Memory Access Time (AMAT) is the average to access memory considering both hits and misses

AMAT = Time for a hit + Miss rate x Miss penalty

Average memory access time (AMAT) (Part 2/2)

- What is the AMAT for a processor with clock frequency 2 Ghz and with miss penalty of 25ns, a miss rate of 2% misses per instruction and a cache access time of 1 clock cycle?

  Miss penalty in clock cycles= access time/clk period

  = 25ns/(0.5 ns/ cycle) =50 clock cycles.

  Time for hit= cache access time =1 clock cycle

  Time for miss= miss rate * miss penalty=2% * 50=1

  AMAT = $t_{avg} = t_{hit}$ + miss-rate * miss penalty= 1 + 0.02x50 = 2

Hence a better metric for cache performance is AMAT

Cache Summary

Four questions

- Placement
  - Direct-mapped, set-associative, fully-associative
- Identification
  - Tag array used for tag check
- Replacement
  - LRU, FIFO, Random
- Write policy
  - Write-through, write-back