CS305
Computer Architecture

Program Performance Analysis in Presence of Cache

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

Reworking the Performance Equation in Presence of Cache

Cache hit ➔ normal operation
CPU time = CPU time without misses + stalls due to misses

Read stalls: # read misses x read miss penalty

Write stalls: depends on write access scheme
For write-back: read misses will potentially have to write dirty blocks
For write-through: write stalls = # write-buffer stalls
For write-through + write-allocate:
write miss penalty = read miss penalty + % write-buffer stalls

Note: miss rate is given in terms of # memory accesses

Program Performance Analysis in Presence of Cache: An Example

I-cache miss rate: 1%, D-cache miss rate: 5%, % memory instructions = 40%
Miss penalty = 100 cycles
CPI without memory stalls = 3, CPI with memory stalls = ?

CPI with memory stalls = 3 + 1% x 100 cycles + 40% x 5% x 100 cycles = 6

Suppose original CPI halved due to better pipelining and data forwarding

New CPI = 1.5 + 3 = 4.5, effective speedup = 6/4.5 = 4/3 only, not 2

Suppose CPI further halved (e.g. superscalar architecture), new speedup?

New CPI = 0.75 + 3 = 3.75, effective speedup = 4.5/3.75 = 6/5 only, not 2

Amdahl's Law

Program Performance with Multi-Level Caches: An Example

L1 miss rate = 2%, L2 miss rate (global) = 0.5% (both w.r.t. executed instructions)
Miss penalty into L2 = 25 cycles, miss penalty into main memory = 500 cycles
CPI without misses = 2.5, what is the performance improvement due to L2 ?

CPI without L2 = 2.5 + 2% x 500 = 12.5
CPI with L2 = 2.5 + 2% x 25 + 0.5% x 500 = 5.5
Performance improvement due to L2 = 12.5/5.5

Summary

- Program performance with cache: **add to CPI** corresponding to miss rate and miss penalty
- Instance of **Amdahl's law**: improvements to processor almost useless beyond a point
- Octa-core processor versus quad-core processor: what do you think is the performance difference?
- Questions most relevant to program performance: **cache configuration** (cache size, number of levels, on-chip versus off-chip)