CSE 420/548 Comp Arch I Sep 13, 2008

- Q2 distributed
- Programming Assignment Discussion
- Review of Memory Hierarchy - App. C
  - Fig. C.1 - different level of memory hierarchy, access times, managed by
  - Cache Performance Review
- Four Qs for Memory Hierarchy
Assignment 1: Cache Simulator

- Maintain the current state of the cache (in a structured array)

- Upon reading an address from address trace
  1. Decide whether hit/miss
  2. Update cache state
  3. Update state

\[
\text{for } i = 1 \text{ to } n \\
\text{Sum} = \text{Sum} + 10 \text{ Sum } \text{ + } \text{A}(c)
\]
Register Allocation by Compilers: Compiler decides Mapping

Register ← Current Value

→ Save $10

XYZ was using $10

→ If Sum is assigned to Reg $10

Sum = Sum + AE; ← registers

→ Use $10

XYZ = Sum + ...

→ Save $10

Use reg $10

→ Load temp mem $10

Overhead (expensive)

Register Allocation tries to minimize this overhead

Remember: Different levels of Memory Hierarchy are maintained by different hardware (e.g., cache controller) or software components (e.g., compiler).
Cache Performance

\[ \text{CPU time} = \frac{IC \times CPI \times \text{clock cycle time}}{\text{CPU clock cycles}} \]

\[ = \left( \frac{\text{CPU clock + Memory stall cycles}}{\text{CPU clock cycles}} \right) \times \text{clock cycle time} \]

Memory stall cycles = Number of misses x Miss penalty

\[ = IC \times \frac{\text{Misses}}{\text{Instruction}} \times \text{Miss penalty} \]

\[ = IC \times \frac{\text{Memory access}}{\text{instruction}} \times \text{Miss rate} \times \text{Miss penalty} \]
II Performance of a computer with memory stalls:

\[
\text{Memory stall cycles} = IC \times \frac{\text{Memory access \times Miss rate \times Miss penalty}}{\text{Instr}}
\]

\[
= IC \times (1 + 0.5) \times 0.02 \times 25
\]

\[
= IC \times 0.75
\]

\[
\text{CPU execute cache} = \left(\frac{IC}{IC + 0.75IC}\right) \times \text{Clock cycle}
\]

\[
= 1.75 \times IC \times \text{Clock cycle}
\]

\[
\frac{\text{Performance CPU}}{\text{Performance CPU cache}} = \frac{\text{CPU execute cache}}{\text{CPU execute CPU}} = \frac{1.75 IC \times \text{Clock cycle}}{1.0 IC \times \text{Clock cycle}}
\]

\[
= 1.75
\]
Example Pg C-5:

Given CPI = 1.0 — when all memory access hit in the cache.

Only data access thru load & store & there are 50% of the instr.
miss penalty = 25 clock cycle
miss rate = 2%.

How much faster would the computer be if all instr were cache hits?

Ans: Performance of a computer that always hits

CPU execution time = (CPU clock cycles + Memory stall cycles) x
= IC x CPI
= IC x 1 x Clock cycle
= IC x 1 x Clock cycle
Four Memory Hierarchy Questions (p. C-6)

Q1: Where can a block be placed in the upper layer (level)? (block placement)

Q2: How is a block found in the upper layer? (block identification)

Q3: Which block should be replaced on a miss? (block replacement)

Q4: What happens on a write? (write strategy)