REVL WALTER MBOGO NJUKI – SCT212-0053/2020

Computer Architecture – tutorial 4[TUTOR COPY]

Context, Objectives and Organization

This worksheet covers the material from lectures on Caches. The goal of the quantitative exercises in this tutorial is to familiarize you with quantitative analysis of caches (E1) and to investigate the tradeoffs between write-through and write-back caches (E2 and E3).

E1: individual – 10 min

Problem

Assume we have a computer where the CPI is 1.0 when all memory accesses (including data and instruction accesses) hit in the cache. The cache is a unified (data + instruction) cache of size 256 KB, 4-way set associative, with a block size of 64 bytes. The data accesses (loads and stores) constitute 50% of the instructions. The unified cache has a miss penalty of 25 clock cycles and a miss rate of 2%. Assume 32 bit instruction and data addresses.

a. What is the tag size for the cache?

b. How much faster would the computer be if all memory accesses were cache hits?

Solution

A.

- **Tag Size**
- **Cache size** = 256 KB = 2^{20} bytes
- **Block size** = 64 bytes = 2^{6}
- **Associativity** = 4-way
- **Address size** = 32 bits

- Number of blocks = 2^{20} / 2^{6} = 2^{14} blocks
- Number of sets = 2^{14} / 4 = 2^{12} = 4096 sets
- Bits for **block offset** = $\log_2(64)$ = **6 bits**

- Bits for **index** = $\log_2(4096)$ = **12 bits**
- Bits for **tag** = 32 − (12 + 6) = **14 bits**

= **14 tag bits**

b. CPI with Cache Misses

- **Ideal CPI** = 1.0
- **Miss rate** = 2% = 0.02
- **Miss penalty** = 25 cycles
- **Instruction access** = 1 per instruction
- **Data access** = 0.5 per instruction (since data accesses are 50% of instructions)

- Total memory accesses per instruction = 1 (instruction) + 0.5 (data) = **1.5**
- Misses per instruction = 1.5 × 0.02 = **0.03**
- Stalls per instruction = 0.03 × 25 = **0.75 cycles**
- Total CPI = 1.0 (base CPI) + 0.75 = **1.75**

Speedup = 1.75 / 1.0 = **1.75×

The computer with no cache misses is **1.75 times faster**

E2: groups of 2 – 15 min

Problem

You purchased an Acme computer with the following features:

- 95% of all memory accesses are found in the cache.
- Each cache block is two words, and the whole block is read on any miss.

- The processor sends references to its cache at the rate of 10^9 words per second.

- 25% of those references are writes.

- Assume that the memory system can support 10^9 words per second, reads or writes.
- The bus reads or writes a single word at a time (the memory system cannot read or write two words at once).
- Assume at any one time, 30% of the blocks in the cache have been modified.
- The cache uses write allocate on a write miss.

You are considering adding a peripheral to the system, and you want to know how much of the memory system bandwidth is already used. Calculate the percentage of memory system bandwidth used on the average in the two cases below. Be sure to state your assumptions.

a. The cache is write through.

b. The cache is write back.

Solution

We know:

- Miss rate = 5% = 0.05
- Block size = 2 words
- Processor memory reference rate = 10^9 references/sec
- Write frequency = 0.25×10^9
- Read frequency = 0.75×10^9
- Bus transfer size = 1 word
- Write allocate is used

Breakdown;

- Read hits = $0.75 \times 0.95 = 0.7125$
- Read misses = $0.75 \times 0.05 = 0.0375$
- Write hits = $0.25 \times 0.95 = 0.2375$
- Write misses = $0.25 \times 0.05 = 0.0125$

a. Write through cache

- **Read hit**: no memory traffic
- **Read miss**: read 2 words from memory
- **Write hit**: write 1 word to memory
- **Write miss**: read 2 words (write-allocate) + write 1 word to memory

Words Transferred Calculation

Words transferred = (0.0375×2) + (0.2375×1) + (0.0125×3)
=0.075+0.2375+0.0375=0.35 words/reference

Average Bandwidth Used:

0.35 × 10^9 = 3.5 × 10^8 words/second

Memory Bandwidth Utilization:

(3.5 × 10^8) × 100 = 35%

b. Write back cache

Then:

Read/write hit: no memory access
Miss (clean evict): read 2 words from memory
Miss (dirty evict): write back 2 words + read 2 words = 4 total
Avg transfer per miss = 0.7 × 2 + 0.3 × 4 = 2.6
Total misses = 0.0375 (reads) + 0.0125 (writes) = 0.05
Total words transferred = 0.05 × 2.6 = 0.13
Average bandwidth used = 0.13 × 10^9 = 1.3 × 10^8 words/sec
Bandwidth % = 0.13 × 100 = 13%

Answer = 13% bandwidth used

Conclusion: Comparing 1 and 2 we notice that the write through cache uses more than twice the cache-memory bandwidth of the write back cache.

E3: groups of 2 – 15 min

Problem

One difference between a write-through cache and a write-back cache can be in the time it takes to write. During the first cycle, we detect whether a hit will occur, and during the second (assuming a hit) we actually write the data. Let's assume that 50% of the blocks are dirty for a write-back cache. For this question, assume that the write buffer for the write through will never stall the CPU (no penalty). Assume a cache read hit takes 1 clock cycle, the cache miss penalty is 50 clock cycles, and a block write from the cache to main memory takes 50 clock cycles. Finally, assume the instruction cache miss rate is 0.5% and the data cache miss rate is 1%. Assuming that on average 26% and 9% of instructions in the workload are loads and stores, respectively, estimate the performance of a write-through cache with a two-cycle write versus a write-back cache with a two-cycle write.

Solution

CPU performance equation: $CPUTime = IC * CPI * ClockTime$

$CPI = CPI_{execution} + StallCyclesPerInstruction$

We know:

Instruction miss penalty is 50 cycles

Data read hit takes 1 cycle

Data write hit takes 2 cycles

Data miss penalty is 50 cycles for write through cache

Data miss penalty is 50 cycles or 100 cycles for write back cache

Miss rate is 1% for data cache (MRD) and 0.5% for instruction cache (MRI)

50% of cache blocks are dirty in the write back cache

26% of all instructions are loads

9% of all instructions are stores

Then:

$CPI_{execution} = 0.26 * 1 + 0.09 * 2 + 0.65 * 1 = 1.09$

Write through

$StallCyclesPerInstruction = MRI * 50 + MRD * (0.26 * 50 + 0.09 * 50) =$ 0.425 so:

$$CPI = 1.09 + 0.425 = 1.515 \tag{3}$$

Write back

$StallCyclesPerInstruction = MRI * 50 + MRD * (0.26 * (0.5 * 50 + 0.5 * 100) + 0.09 *$
$(0.5 * 50 + 0.5 * 100)) =$
0.5125 so:

$$CPI = 1.09 + 0.5125 = 1.6025 \tag{4}$$

Comparing 3 and 4 we notice that the system with the write back cache is 6% slower.

Note: The question assumes that for the write back cache, there is no write back buffer. If there was a write back buffer then the two caches will behave identically.

aVijay Nagarajan 2017. Thanks to Boris Grot, Nigel Topham and Marcelo Cintra