Computer Architecture – tutorial 4[SOLUTIONS]

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

Number of bits used for block offset = log 64 = 6.
Number of sets in the cache = 256K/(64 * 4) = 1K
Number of bits for index = log 1K = 10
Number of bits for tag = 32 - (10 + 6) = 16

Now,
$CPI = CPI_{execution} + StallCyclesPerInstruction$
For computer that always hits, $CPI = 1$
For computer with non-zero miss rate, let us compute $StallCyclesPerInstruction$
$StallCyclesPerInstruction$ = (Memory accesses per instr) * miss rate * miss penalty
Memory accesses per instruction = 1 + 0.5 (1 instruction access + 0.5 data access)
$StallCyclesPerInstruction$ = 1.5 * 0.02 * 25 = 0.75
Therefore, $CPI = 1.75$
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 = 0.05
Block size = 2 words (8 bytes)
Frequency of memory operations from processor = 10^9
Frequency of writes from processor = $0.25 * 10^9$
Bus can only transfer one word at a time to/from processor/memory
On average 30% of blocks in the cache have been modified (must be written back in the case of the write back cache)
Cache is write allocate
So:
Fraction of read hits = $0.75 * 0.95 = 0.7125$
Fraction of read misses = $0.75 * 0.05 = 0.0375$
Fraction of write hits = $0.25 * 0.95 = 0.2375$
Fraction of write misses = $0.25 * 0.05 = 0.0125$

a. Write through cache

Then:
On a read hit there is no memory access
On a read miss memory must send two words to the cache
On a write hit the cache must send a word to memory

On a write miss memory must send two words to the cache, and then the cache must send a word to memory

Thus:

Average words transferred $= 0.7125 * 0 + 0.0375 * 2 + 0.2375 * 1 + 0.0125 * 3 = 0.35$

Average bandwidth used $= 0.35 * 10^9$

$$Fractionofbandwidthused = \frac{0.35 * 10^9}{10^9} = 0.35 \quad (1)$$

b. Write back cache

Then:

On a read hit there is no memory access

On a read miss:

1. If replaced line is modified then cache must send two words to memory, and then memory must send two words to the cache

2. If replaced line is clean then memory must send two words to the cache

On a write hit there is no memory access

On a write miss:

1. If replaced line is modified then cache must send two words to memory, and then memory must send two words to the cache

2. If replaced line is clean then memory must send two words to the cache

Thus:

Average words transferred $= 0.7125 * 0 + 0.0375 * (0.7 * 2 + 0.3 * 4) + 0.2375 * 0 + 0.0125 * (0.7 * 2 + 0.3 * 4) = 0.13$ Average bandwidth used $= 0.13 * 10^9$

$$Fractionofbandwidthused = \frac{0.13 * 10^9}{10^9} = 0.13 \quad (2)$$

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.

Boris Grot 2018. Thanks to Vijay Nagarajan, Nigel Topham and Marcelo Cintra