Student ID: 40055665

Student Name: Chao Ye

COMP 5201 Assignment 4 Fall 2017

Issued: Tuesday, November 21, 2017 Tuesday, December 5, 2017

Submit typed hardcopy as instructed. No extension will be granted.

1. [20 marks] We investigate the increase in CPI (clocks per instruction) due to cache misses that occur during memory references. For simplicity, we pretend that instruction fetches never miss.

a) [10 marks] Suppose the processor takes an average of 1.5 clock cycles to execute an instruction when there are no cache misses. Assume that the miss penalty is 8 cycles and that there is an average of 1 memory reference per 3 instructions.

This \_base\_ CPI of 1.5 cycles includes the cache hit time.

Suppose the miss rate is 5%. Using the formula t\_ave = ht + mr \* mp, what is the CPI when cache misses are taken into account? <average time = hit time + miss rate \* miss penalty>

**Cache miss time = miss rate \* miss penalty**

**Cache miss time = 0.05 \* 8 = 0.4 clock cycles**

**Because every 3 instructions have one memory reference**

**So 0.4/3 clock cycles per instruction**

**Execute an instruction take 1.5 clock cycles**

**CPI = 0.4/3 + 1.5 ≈ 1.63 clock cycles per instruction**

b) [10 marks] Consider the same processor with a two-level cache. The hit rates for the L1$ and the L2$ are 95% and 80%, respectively. The \_local\_ miss penalties are 8 cycles and 60 cycles, respectively. Assume the same density of memory references.

If the CPI is 1.5 cycles when there are no cache misses, what is the CPI when cache misses are taken into account?

Hint: Apply the formula recursively to find the effective miss penalty of the L1$.

**Cache miss time = L1 miss rate \* L1 miss penalty + L1 & L2 miss rate \* L2 miss penalty**

**Cache miss time = 0.05 \* 8 + 0.05 \* 0.2 \* 60 = 1**

**Because every 3 instructions have one memory reference**

**So 1/3 clock cycles per instruction**

**Execute an instruction take 1.5 clock cycles**

**CPI = 1/3 + 1.5 ≈ 1.83 clock cycles per instruction**

2. [24 marks] 1) \_Compulsory\_ misses occur the first time a program touches a cache line. 2) \_Conflict\_ misses occur when more than 'm' lines map to the same set in an m-way set-associative cache. 3) \_Capacity\_ misses occur when a program's working set exceeds the cache capacity.

It is hard to do much about compulsory misses, but both conflict and capacity misses are affected by the geometry parameters of a cache: its capacity, its associativity (wayness), and its cache-line size.

a) [8 marks] If we increase the capacity 'S' of the cache, but keep the other two parameters constant, will i) conflict misses, and ii) capacity misses, increase or decrease? Explain.

Also, is there a downside to larger caches? Explain.

**Capacity misses will decrease because increase the capacity ‘S’ of the cache will allow more cache lines store into the cache at the same time.**

**Conflict misses will decrease because the conflict of two addresses is less likely to happen.**

**Larger caches cost more.**

b) [8 marks] If we increase the associativity (wayness) 'm' of the cache, but keep the other two parameters constant, will i) conflict misses, and ii) capacity misses, increase or decrease? Explain. Also, is there a downside to increased associativity? Explain.

**Capacity misses will not change because the total capacity is not change by increase the associativity. In fact, a capacity miss requires a fully associative cache.**

**Conflict misses will decrease because the value of ‘m’ is bigger means that more cache lines can be compute for the same cache frames.**

**Increased associativity may enlarge the latency (using more time and power).**

c) [8 marks] If we increase the cache-line size 'L' of the cache, but keep the other two parameters constant, will the miss rate increase or decrease i) for programs with a high spatial locality, and ii) for programs with low spatial locality? Explain.

**For programs with a high spatial locality, the miss rate will decrease because the programs will use relative data within the cache line.**

**For programs with a low spatial locality, the miss rate will increase because it will not use relative data within the cache line and it will collect a larger cache-line.**

3. [21 marks] Consider a computer with a byte-addressable memory.

A 40-bit memory address is divided as follows for cache processing. First, the 8 low-order bits are chopped off to expose the cache-line number.

Second, the next 17 low-order bits are inspected to get the cache-container index.

Third, the remaining 15 bits are used as the cache tag.

Hint: What do the direct-mapped and set-associative placement formulas have in common?

1. [7 marks] What is the cache size in bytes?

**Cache size include cache line number and cache container index**

**8 + 17 = 25**

**Cache size is 225 bytes**

1. [7 marks] What is the cache-mapping scheme?

**Direct - mapping**

1. [7 marks] For a given byte in the cache, how many different bytes in the main memory could possibly be mapped to it?

**215 bytes**

4. [19 marks] Consider a computer with 32-bit registers. The memory is word addressed. There is a direct-mapped cache with 4K cache frames.

Cache lines are 16 words. Into which cache frame, and with what tag value, does 32-bit word address '45677cba' go? Show your work. Answers in hexadecimal.

**Cache lines number = byte address div number of bytes/line**

**Cache lines number = 45677cba(16) div 10(16) = 45677cb(16)**

**Frames number = cache lines number mod number of frames/cache**

**Frames number = 45677cb(16) mod 163 = 7cb(16)**

**Tag value = 4567(16)**

5. [16 marks] Consider a word-addressed computer whose memory latency is 200 cycles.

The hardware bandwidth is 1 word/cycle.

The processor manages to sustain 180 outstanding (load) memory references in each and every cycle.

a) [8 marks] Using Little's law, calculate the (potential) software bandwidth in words/cycle arriving at the processor.

Does the hardware bandwidth limit you? Show your work.

**The software bandwidth is depend on the minimum number between memory-reference concurrency divided by the memory latency and the hardware bandwidth.**

**180/200 = 0.9 > 1**

**So the software bandwidth is 0.9 words/cycle**

**The hardware bandwidth does not limit the software bandwidth.**

b) [8 marks] The processor has a cache, with a 1-word cache line, whose hit rate is 95%. Assuming that each arithmetic operation requires one new operand, and that the \_peak\_ arithmetic performance of the processor is 20 operations/cycle, what is the \_sustained\_ arithmetic performance of this processor in operations per cycle? Show your work.

**Bandwidth is 0.9 words/cycle**

**Cache line is 1-word**

**Peak arithmetic performance is 20 operations/cycle**

**Hit rate is 95%**

**So sustained arithmetic performance of this processor is**

**0.9/1 = 0.9**

**0.9/(1 - 0.95) = 18 operations/cycle < 20**

From probst at cse.concordia.ca Wed Nov 22 16:49:38 2017

From: probst at cse.concordia.ca (David K. Probst)

Date: Wed, 22 Nov 2017 16:49:38 -0500

Subject: [comp5201-f17] from last lecture (pipeline magic,

too advanced for COMP5201)

Message-ID: <5a15f0f2.iiVoyWPTTDeie8nD%probst@cse.concordia.ca>