Assignment 1

Kyle Cheung

Feburary 4, 2018

1

1.1

First we form a system of equations by multiplying each instruction by how many clock cycles it uses. We set Load instruction clock cycles to be y, store clock cycles to z and the speed of program 1 to x. We know that program 2 and 3 are 2 and 3 times slower respectively, thus

$$x = 500y + 200z + 850$$
$$2x = 1330y + 200z + 850$$
$$3x = 500y + 2275z + 850$$

Solving this system of equations yields that Load instruction uses 5 clock-cycles and Store instruction uses 4 clock-cycles

1.2

$$CPI = \frac{700}{1450} * 1 + \frac{500}{1450} * 5 + \frac{200}{1450} * 4 + \frac{50}{1450} * 3$$

$$CPI = 2.86$$

$$ExecutionTime = \frac{InstructionCount * CPI}{ClockRate}$$

$$ExecutionTime = \frac{1450 * 2.86}{3,000,000,000}$$

$$ExecutionTime = 0.00000138333$$

1.3

$$LoadStoreInstructions = \frac{700}{1450}$$

Memory Stall Cycle = 1 * Instructions cache Miss Rate * Miss Penalty + Load Store Instructions * Data-cache miss rate * Miss Penalty

$$MemoryStallCycle = 1 * .02 * 100 + \frac{14}{29} * .03 * 100$$

$$MemoryStallCycle = 3.4483$$

$$CPU - Time = \frac{TotalInstructionCount * (CPI + MissCycle)}{3,000,000,000}$$

therefore exection time including memory stalls is 0.00000305

2

2.1

$$GlobalMissRate = .05 * .08 * .10$$

$$GlobalMissRate = .0004$$

2.2

AMAT = time for hit + miss rate * miss penalty

$$AMAT = 1 + (1 - .95)(4 + (1 - .92)(8 + (1 - .9)(15)))$$

$$AMAT = 1.238$$

2.3

Since the hit rate of L3 is 100% we can remove the miss penalty from it in the AMAT formula

$$AMAT = 1 + (1 - .95)(4 + (1 - .92)(8))$$

$$AMAT = 1.232$$

$$CPI - stall = 1.232 * .78 + 3$$

$$CPI - stall = 3.961$$

3

3.1

total cache misses = $\frac{2^{20}}{16}$
total cache hits = $2^{20} * 2 - \frac{2^{20}}{16}$
This program has spatial locality, it's cache misses are cold and capacity.

3.2

total cache misses = $\frac{2^{12}}{16}$
total cache hits = $2^{20} * 2 - \frac{2^{12}}{16}$
This program has spatial and temporal locality, it has cold cache misses.

3.3

total cache misses $= \frac{2^{20}}{16}$
total cache hits $= 2^{20} * 2 - \frac{2^{20}}{16}$
This program has spatial locality, it has cold and capacity cache misses.

3.4

total cache misses $= 2^{19} * 4$
total cache hits $= 0$
This program has no locality displayed, it has cold and capicty cache misses

4

4.1

For my computer the best block values I had were
BLOCK_X = 64
BLOCK_Y = 32
BLOCK_Z = 1024

4.2

The two programs allocate memory then create and multiple matrices. The Naive version does a brute force method that just loops straight through the values and multiplys them. The tiled version creates blocks based on the block sizes and iterates through them calling them into cache. Then the blocks still in cache are operated on first, taking advantage of locality and cache.

4.3

4 cores

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 142
model name      : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz
stepping        : 9
microcode       : 0x42
cpu MHz         : 698.821
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 0
```

```
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse3
bugs            : cpu_meltdown
bogomips        : 5426.00
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual
power management:
```

4.4

The tile version is faster. The tiled version uses cache blocks to reduce memory access time while the naive version just iterates over the loop.

5

5.1

Address	Hit/Miss
4	miss
194	miss
46	miss
6	miss
196	miss
94	miss
197	hit
22	miss
6	miss
54	miss
195	hit
265	miss

5.2

Address	Hit/Miss
4	miss
194	miss
46	miss
6	miss
196	miss
94	miss
197	hit
22	miss
6	hit
54	miss
195	hit
265	miss