CMSC 22200 Computer Architecture
The University of Chicago Autumn 2015
Lab Assignment 3
Due: Tuesday, December 1 at the beginning of class by hardcopy

In this lab assignment, we explore cache configurations, their latencies, bandwidth and energy consumption using the Marss simulator.

Part 1. Exploring Cache Configurations

In this part, we study the statistics produced by the simulator for different cache configurations. We intend to explore the impact of various cache design choices such as capacity, block size, and associativity.

1. Go to your marss directory in your CS home directory. Rename the "config" directory to "oldconfig". Then, download the configuration archive from the course web site onto your CSIL Linux machine. Unzip it, yielding a new "config" directory. If it isn't already there, place this new config directory into your marss directory (where the old config directory was). Then recompile marss by entering the following commands:

```
scons -Q -c
scons -Q
```

2. Open the marss/config folder and open the default.conf file. You will find several different machine configurations, and each is named with the same format. For example:

64k_64b_8way is a machine that has an L1 cache which has a size of 64KB, a cache line size of 64 bytes and is 8-way associative.

If you want to assign a different machine, set the machine name as below:

```
Simconfig -machine cache_64k_64b_8way -stats cache_ 64k_64b_8way.st
```

(to the above line, you would also add the logfile and stopinsns flags and parameters)

Now you can profile the Parsec benchmarks (Blackscholes and Swaptions) based on the various cache configurations. Similar to previous labs, run each benchmark for 200 million instructions.

3. Run each of the Parsec benchmarks on the basecache configuration (cache_64k_64b_8way) and answer the following questions based on the

statistics. (Note that the statistics files contain stats on the L1 data cache and the L1 instruction cache; please take a moment to become acquainted with the layout of the information so you know where to find the necessary information. Unless otherwise directed, sum data and instruction cache statistics.)

Q1. What is the number of L1 cache references? Report the total number for each of the benchmarks, in each case summing data and instruction caches.

Q2. What fraction of the executed instructions are loads? What fraction are stores? What fraction are branches? (Use the opclass section under issue and divide by the total number of uops in the section above it, not by the total number of instructions.)

Q3. What is the CPI?

Q4. While it is difficult to clearly calculate the CPI contribution of the memory hierarchy in a sophisticated modern architecture, we can approximate it with the miss rate and miss time. Using the miss rate of the L1 cache, and a miss penalty of 25 clock periods, what is the contribution of the memory hierarchy to the CPI? In other words, how many cycles per instruction are spent servicing misses?

4. Run each of the benchmarks on three different L1 capacity configurations: (cache_64k_64b_8way, cache_128k_64b_8way, cache_256k_64b_8way).

Q5. What are the L1 hit rates for each of the cache configurations? Explain why they vary.

Q6. Is the variation large?

5. Run each of the benchmarks on three different L1 cache organizations, with different block sizes (cache_64k_64b_8way, cache_64k_128b_8way, cache_64k_256b_8way). This explores the phenomena of spatial locality.

Q7. What are the L1 hit rates for each of the cache configurations? Explain why they vary.

Q8. Is the variation large?

6. Now we explore different associativity structures, which is the flexibility in where data can be placed in the cache. Run each of the benchmarks on various associativity cache organizations: (cache_64k_64b_directmapped, cache_64k_64b_2way, cache_64k_64b_4way, cache_64k_64b_fullyassociative). Note that fully associative cannot be practically implemented, but it is studied as a measure of potential cache performance in the extreme.

Q9. What are the L1 hit rates for each of the cache configurations? Explain why they vary.

Q10. Is the variation large?

6. One of the dimensions of cache performance that is underappreciated is the critical role of cache memories in reducing the data bandwidth needed as we move away from the processor. This reduction is critical to why cache memories work, and our ability to sustain increased performance. While the needs of both instruction fetch and data access are significant, here we will focus only on the data accesses.

Q11. Look back to your runs on the base cache configuration (which we will be referring to exclusively for the rest of the assignment). Based on the runtime (in cycles) of the program and the instruction mix, at the ISA level, what is the number of bytes needed from the memory hierarchy to execute this program? (To make it simple, we only look at data memory, and you can assume each read or write is one 32-bit word or 4 bytes.)

Q12. How many bytes per instruction is this? How many bytes per cycle is this? Assuming a 2Ghz clock, what is the required rate in bytes / second?

7. Now consider the data bandwidth required downstream from the L1 cache, that is, accesses to L2 cache. We can compute this by looking at the number of L1 cache misses (for the L1 data cache only, as we continue to ingnore instruction fetch), and data transferred per miss. Here we simplify it by assuming it is exactly 64bytes/miss, though in practice the number is larger because the victim cache line may be dirty and needs to be written back.

Q13. How many bytes are needed from the L2 cache to execute this program? What reduction did the L1 cache achieve (give a ratio)?

8. Finally, repeat the steps for the DRAM system, assuming you can do this by looking at L2 cache misses and the 64B block size.

Q14. How many bytes are needed from the memory to execute this program? What reduction did the L2 cache achieve (give a ratio)?

(Note that you have now characterized the bandwidth needed out of the L1 cache, out of the L2 cache, and out of main memory, and can take a look back at and compare them all.)

9. Power is an increasingly critical limit in computing systems. In this problem, we use what you’ve learned to explore how cache memories affect power and performance.

Q15. Assuming the following access energies L1 access (1 picojoule/byte or 4 picojoules per word), and L2 access (4 picojoules/byte), and for DRAM (500 nanojoules/64B block), How much energy is consumed at each level of the memory hierarchy for the accesses to execute each of the benchmarks? (To keep it simple, just count the energy for accesses at each level as above, don’t worry about other energy needed to move entire cache blocks)

Q16. How much energy would it take to execute each of the benchmarks, if all of the program’s memory references went to DRAM? Using a 2GHz clock, what would the power consumption be? Is this feasible? To put this number in perspective, identify an every-day object that has a similar scale of power consumption to this.